ERDC/GSL TR-12-36

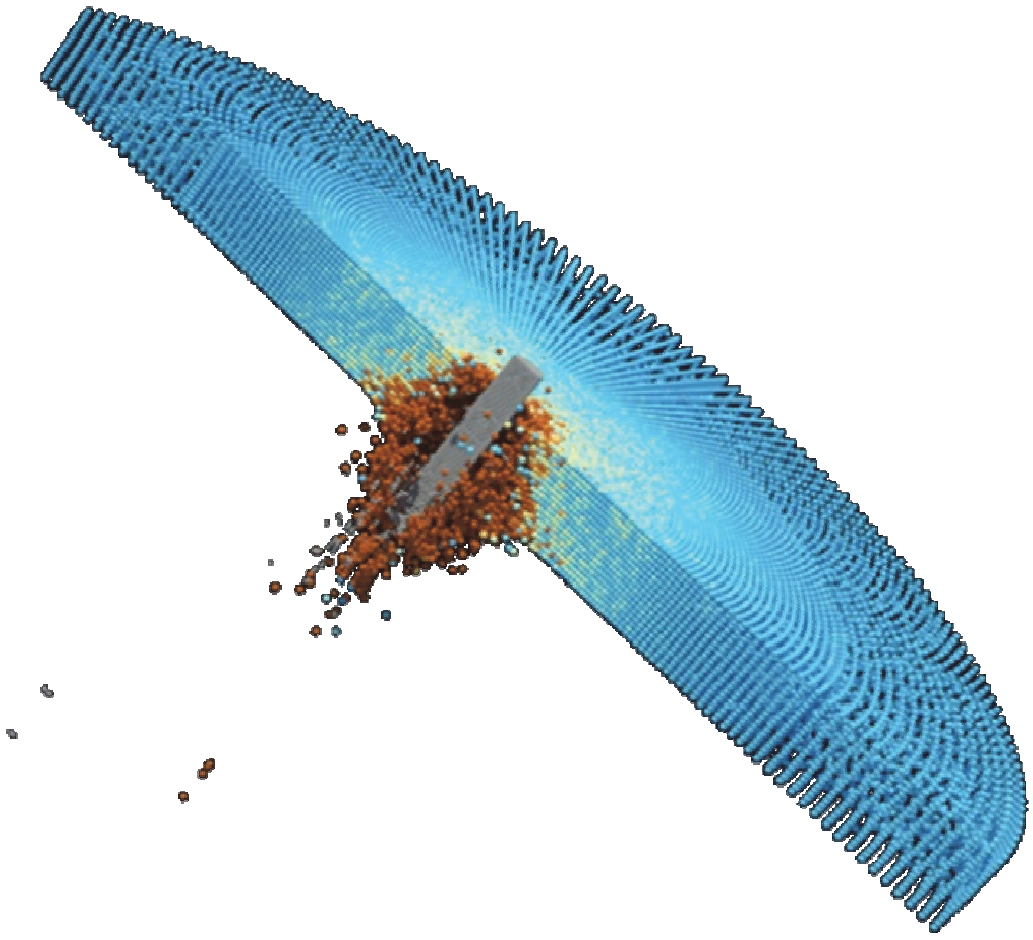
Geotechnical and Structures Laboratory

US Army Corps
of Engineers®
Engineer Research and
Development Center

User's Manual for Nonlinear Meshfree Analysis Program (NMAP) Version 1.0

S. W. Chi, C. H. Lee, J. S. Chen, M. Jason Roth,
and Thomas R. Slawson

December 2012



ERDC/GSL TR-12-36
December 2012

User's Manual for Nonlinear Meshfree Analysis Program (NMAP) Version 1.0

M. Jason Roth and Thomas R. Slawson

Geotechnical and Structures Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

S. W. Chi, C. H. Lee and J. S. Chen

Department of Civil and Environmental Engineering
University of California, Los Angeles
Los Angeles, CA 90095

Final report

Prepared for U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Abstract

As a part of the meshfree method development for fragment-impact modeling, an enhanced semi-Lagrangian reproducing kernel particle method formulation for modeling large material deformation and damage mechanisms was developed. The formulation includes a microcrack-informed damage model (MIDM) for improved material failure modeling. The MIDM is based on multiscale homogenization using the energy bridging theory pertinent to fragment penetration modeling of concrete materials. Enhanced kernel contact algorithms to model multi-body contact applicable to penetration problems were also developed. The computational formulations and numerical algorithms for these new meshfree method developments were implemented into the parallel Nonlinear Meshfree Analysis Program (NMAP) code. This report provides user instructions for building and running a model in NMAP. The report, together with previously published technical reports, also provides a general overview of the theoretical foundation of the newly developed meshfree formulation for fragment-impact modeling.

Contents

Figures and Tables

Figures

Tables

Preface

This report documents the theoretical basis and provides instructions for use of the Nonlinear Meshfree Analysis Program developed by personnel of the University of California, Los Angeles (UCLA) under contract number W912HZ-07-C-0019. The research and code development were performed in collaboration with and directed by personnel of the Survivability Engineering Branch (SvEB), Geosciences and Structures Division (GSD), Geotechnical and Structures Laboratory (GSL), U.S. Army Engineer Research and Development Center (ERDC). Numerical simulations were jointly performed on systems of the ERDC Department of Defense Supercomputing Resource Center and the UCLA computational cluster.

Research funding was provided by the Army Technology Objective-Demonstration research program D.FP.2009.05, DEFeat of Emerging Adaptive Threats (DEFEAT). R. Nicholas Boone, SvEB, was DEFEAT work package manager.

Principal investigator for the UCLA research team was Professor J. S. Chen, Chair and Chancellor's Professor, Department of Civil and Environmental Engineering. ERDC principal investigating engineers were Dr. Thomas R. Slawson and M. Jason Roth, SvEB.

This report was prepared by Chen, S. W. Chi, and C. H. Lee, UCLA, and by Slawson and Roth, SvEB. The research was accomplished under the general supervision of James L. Davis, Chief, SvEB; Bartley P. Durst, Chief, GSD; Dr. William P. Grogan, Deputy Director, GSL; and Dr. David W. Pittman, Director, GSL.

COL Kevin J. Wilson was Commander and Executive Director of ERDC. Dr. Jeffery P. Holland was Director.

1 Introduction

Background and scope

Meshfree methods — such as Smoothed Particle Hydrodynamics (Gingold and Monaghan 1977); the Element Free Galerkin Method (Belytschko, Lu, and Gu 1994) based on Moving Least Squares Approximation (Lancaster and Salkauskas 1981); and the Reproducing Kernel Particle Method (Liu, Jun, and Zhang 1995; Chen et al. 1996) — represent a relatively new class of numerical methods that are capable of providing distinct advantages over the widely used Finite Element Method (FEM). It is well known that, although FEM has been applied to a wide range of engineering and scientific problems, the method is also known to suffer from significant difficulties in certain areas. These include, for example, problems with moving discontinuities and numerical instability under the presence of excessive element distortion or entanglement in large deformation problems.

Recent research (Chen et al. 1996; Chen and Wang 2000) showed that the newly developed meshfree methods, such as the Reproducing Kernel Particle Method (RKPM), are capable of overcoming the aforementioned numerical shortcomings. Specifically, these methods provide significantly enhanced capabilities for problems involving large deformations and material fragmentation (Chen et al. 1996, 2009; Guan et al. 2009, 2011). In meshfree methods, the interaction between nodes is determined by the overlapping support zones (without the presence of a structured mesh), in contrast to the mesh connectivity required for FEM. Association of the meshfree nodes in this manner relaxes model dependence on a prescribed mesh structure and, therefore, allows the method to effectively model material fragmentation and perform adaptive refinement, among other advantages.

This manual provides documentation for usage of the RKPM-based code *Nonlinear Meshfree Analysis Program (NMAP), Version 1.0*. The *NMAP* code was developed by the research group of J. S. Chen, Civil and Environmental Engineering Department, University of California, Los Angeles. The *NMAP* code was developed for the purpose of analyzing dynamic, nonlinear solid and structural mechanics problems and was enhanced for modeling high-strain-rate, contact-impact problems as a part of contract

W912HZ-07-C-0019, sponsored by the U.S. Army Engineer Research and Development Center (ERDC). A particular enhancement under the current project is a modified Advanced Fundamental Concrete (AFC) model for simulation of projectile penetration into a brittle medium. Additional implemented capabilities essential to fragment-impact modeling include (1) kernel contact algorithms for multi-body interaction, (2) algorithms for adaptive refinement in critical zones, (3) development of a "micro-cracks informed damage model" based on an energy-equivalent bridging (Ren et al. 2011), and (4) implementation of the micro-cracks informed damage model into the AFC model (Adley et al. 2010).

Chapter 1 gives an introduction to the *NMAP* code and an overview of key capabilities included in *Version 1.0*. Chapter 2 provides an overview of the model development process, including preprocessing and data output. Chapter 3 presents a detailed description of the control parameters and input file requirements. Chapter 4 discusses the code structure, and Chapter 5 provides a brief introduction of the theoretical aspects of the numerical formulation. Finally, Chapter 6 presents example problems that were used for code verification and validation.

Overview, formulation, and code capabilities

NMAP, Version 1.0 is a three-dimensional (3-D), explicit RKPM-based code developed for the dynamic analysis of linear and nonlinear solid mechanics problems. The code is based on Lagrangian and semi-Lagrangian RKPM formulations with stabilized nodal integration techniques for domain integration. The RKPM formulation is primarily used as the modeling technique, although the capability to perform RKPM-FEM coupling is also included in the code. An updated Lagrangian framework is used to model geometric and material nonlinearity, and objective stress calculations are ensured by use of the Hughes-Winget algorithm for stress update (Hughes and Winget 1980).

Spatial integration is performed using either Stabilized Conforming Nodal Integration (SCNI) (Chen et al. 2001; Chen, Yoon, and Wu 2002) or Stabilized Nonconforming Nodal Integration (SNNI) (Chen et al. 2006) and can be selected by the user based on specific problem requirements. Additional stabilization based on Chen and Wu (2007) and Puso, Zywicz, and Chen (2006) is currently underway. In the SCNI approach, conforming integration zones (or smoothing zones) are constructed from a Voronoi diagram at the beginning of the calculation. In the SNNI approach, cuboid

integration zones are adopted. In general, SNNI is necessary for problems in which material damage and fragmentation occur, while SCNI can be employed for problems subjected to moderate deformation for desired accuracy.

Time integration is performed using the Newmark time integration scheme. It is recommended that the central difference method ($\beta = 0$ and $\gamma = 0.5$) be used for penetration problems. The central difference method exhibits conditional stability; thus, the Courant condition for time-step selection is necessary. While the central difference method exhibits second order accuracy, users may select larger values of $\gamma (> 0.5)$ if algorithmic damping is desired, but accuracy will be reduced to first order.

In the *NMAP* code, the reproducing kernel approximation is introduced for both Lagrangian (Chen et al. 1996) and semi-Lagrangian approaches (Chen and Wu 2007; Guan et al. 2009, 2011). In the Lagrangian approach, the reproducing kernel approximation is formulated based on the undeformed configuration, and approximation functions are calculated only at the beginning of the simulation. As such, nodes contained within any given node's support remain unchanged throughout the calculation; and, therefore, the support zones deform with the material. In the *NMAP* code, the Lagrangian approach is coupled with the SCNI integration technique. The benefit in using the Lagrangian approximation with SCNI is that the approximation functions and integration zones can be formulated on the initial configuration (i.e., the conforming smoothing zones are calculated once based on the initial configuration, and the Lagrangian approximation functions are also formed once using the initial configuration). This provides computational efficiency and accuracy for problems with moderate deformation. In contrast, for problems that exhibit severe material distortion and fragmentation, the deformation gradient needed for mapping between deformed and undeformed configurations in the Lagrangian formulation can become non-positive definite and lead to divergence of the numerical solution. Under such a condition, a semi-Lagrangian formulation was shown to be a necessary alternative to the Lagrangian formulation (Chen and Wu 2007; Guan et al. 2009, 2011). In the semi-Lagrangian approach, the reproducing kernel approximation is incrementally updated based on the current or deformed configuration. In this case, neighbor nodes are updated based on the current configuration, and the approximation functions are updated based on the new neighboring node definitions. In *NMAP*, the semi-Lagrangian approach is coupled with the

SNNI integration technique, since it is computationally costly to update the Voronio cells needed for SCNI. In general, the semi-Lagrangian RKPM with SNNI approach is recommended for problems such as high-velocity impact and penetration where severe material distortion and/or fragmentation are expected.

The *NMAP* code currently uses a frictional kernel contact algorithm coupled with a relative velocity-based release algorithm to model the interaction between contacting bodies. In this approach, the explicit definition of contact surfaces is not required (in contrast to conventional contact algorithms used in FEM). Rather, the kernel contact algorithm detects contact based on interaction of kernel functions between bodies. The interaction forces generated from the kernel interaction constitute the contact force. For frictional contact with stick-and-slip conditions, the interaction forces between pairs of particles from different bodies are projected onto the normal and tangential directions of the contact surface. The tangential component is corrected to be proportional to the normal component following the Coulomb friction law. Because contact surfaces are not explicitly defined, a level-set based method was implemented to implicitly represent the contact surface and estimate its normal and tangential directions. This allows the frictional contact to be formulated without imposing the conventional kinematic constraints in the normal and tangential directions. In conjunction with the frictional kernel contact algorithm, a relative velocity-based release algorithm has been implemented in the *NMAP* code. The purpose of the release algorithm is to determine whether two nodes contained in separate bodies are separating. In the event the two nodes are separating from each other, the interaction forces are neglected.

Five material models are currently active in the code and can be assigned to different node groups in the input file. These models include a linear elastic material model, a Drucker-Prager plasticity model, a rate-independent plasticity model with kinematic or isotropic hardening, and two versions of the AFC model (Adley et al. 2010). The first version of the AFC model implemented into *NMAP* is an essentially unmodified version of the initial AFC model developed by ERDC (referred to as the macroscale AFC model). In the macroscale AFC model, all material damage is described by a single shear damage evolution parameter that is used to evolve the yield surface under both tension and compression. This single damage evolution parameter is calculated based on deviatoric plastic

strain, volumetric plastic strain, and the material's ultimate tensile strength. The second version of the AFC model (termed the multiscale AFC model) was enhanced with a microcrack-informed damage evolution function to replace the phenomenological damage evolution function originally implemented in the AFC model. In the current implementation, the tension damage evolution function is constructed based on an energy-bridging homogenization method in conjunction with the mesoscale modeling of mode I crack propagation in a representative volume of a concrete microstructure. A similar approach is in development for the enhancement of shear damage evolution in the AFC model based on the homogenization of mode II crack propagation at the mesoscale.

Although the code is fully meshfree (and, thus, does not require a structured mesh), it is most efficient to generate the nodal discretization using a standard FEM preprocessor. In this way, the model geometry, discretization, and essential boundary conditions can be quickly constructed and exported to a neutral file. A preprocessor (*NMAP_PREPRO*) was developed by UCLA to read a *Patran* (Patran 2010) neutral file and generate the necessary input files for use in *NMAP*. Either 4-noded tetrahedral or 8-noded hexahedral elements may be used in the model generation. However, in the current version, the two element types cannot be jointly used in the generation of a single model.

Code parallelization was performed using the Message Passing Interface technique. In the current version, parallelization is limited to neighboring node searches (for initial calculation and update of the approximation functions) and calculation of the internal force vector.

2 Getting Started

Preprocessor

Two data sets are required to perform the necessary preprocessing for an *NMAP* analysis. These data sets are provided in two files: a neutral file created with a finite element preprocessing program and a user-created control file (*NMAP_Input.dat*). The neutral file is used to define model geometry, discretization, and boundary conditions. The control file is used to define model control parameters, body definitions, material properties and initial conditions. These two user-prepared data sets are read by the UCLA-developed preprocessing program, *NMAP_PREPRO*, which converts the specified data into a series of input files used by the main *NMAP* program. Details of the two data sets and the *NMAP_PREPRO* generated input files are discussed in the following paragraphs.

Although *NMAP, Version 1.0,* is based on a meshfree formulation, initial model geometry, discretization, and boundary conditions are most easily created using a standard finite element (FE) preprocessing program. The FE preprocessor can be used to build the model and generate a neutral file that is read by the preprocessor. Because of licensed availability at UCLA, *Patran* was used during code development. Therefore, it is assumed in *NMAP_PREPRO* that the model data are provided in *Patran* neutral file format. Because the data are provided in a neutral file format, it is possible that other FE preprocessing codes can also be used. However, users should verify consistency between the *Patran* neutral file and neutral files generated by other codes.

With regard to the FE model discretization, *NMAP_PREPRO* can be used to transform models constructed with either 4-node tetrahedral elements or 8-node hexahedral elements. However, these element types cannot be jointly used within a single model. An overview of constructing a *Patran* model for use by *NMAP-PREPRO* is discussed in the following paragraphs.

Model geometry and discretization

The FE pre-processor can be used to define any type of model geometry as necessitated by a given problem statement. In *Patran*, model geometry is defined using the Geometry function. Since the *NMAP* code is used for

analyses of 3-D problems, geometry should be defined using 3-D solid objects. Once model geometry is defined, the model can be discretized using the Element function. Various options are available to construct the mesh, but regardless of the method used, the discretization will provide all information necessary to define RKPM nodal coordinates and integration zones. The model mesh can be constructed using either 8-node hexahedral or 4-node tetrahedral elements, depending on problem requirements. However, once an element type is selected, it must be used for the entire model. Although any type of model discretization (subject to the constraints previously described) may be used, users should follow appropriate meshing practices to obtain the most accurate results from the *NMAP* analysis (i.e., utilize sufficient discretization refinement, avoid abrupt changes in nodal spacing, etc.).

Boundary conditions

Dirichlet and Neumann boundary conditions can be specified in *Patran* using the Loads/BCs function. With respect to the Dirichlet (or essential) boundaries, both zero and non-zero magnitudes may be prescribed. Prescribed displacement boundary conditions with zero magnitude are treated as fixed boundaries in *NMAP* and therefore remain fixed throughout the analysis. However, non-zero displacements are assumed to prescribe total displacements at the specified node. As such, the incremental displacement for each time-step is calculated as the user-specified total displacement divided by the number of time increments used in the analysis. In addition to prescribing total nodal displacements in the neutral file, users are also given the option to apply a uniform scaling factor to the specified displacements in the control parameter file (reference DCURVE in *NMAP_Input.dat)*.

Similar to Dirichlet boundary conditions, Neumann (or natural) boundary conditions may also be specified in *Patran*. Natural boundary conditions are specified as prescribed nodal forces and are applied in the NMAP analysis as constant, total nodal forces at the specified nodes. As they were with the prescribed displacements, users are given the option to scale the prescribed forces with a uniform scaling factor defined in the control parameter input file (reference FCURVE in *NMAP_Input.dat*).

The Dirichlet and Neumann boundary conditions may be separately applied (and may be of different type and magnitude) for all three degrees

of freedom. For all nodes without specified Dirichlet boundary conditions, the nodes are assumed as free degrees of freedom in the *NMAP* analysis.

As an example of constructing initial model geometry and boundary conditions using *Patran*, a simple cantilevered beam model is shown in Figure 1. The beam is fixed at the left end and is subjected to spatially varying nodal forces along the top (magnitude of 1.0 on the left half of the beam and magnitude of 2.0 on the right half). If in the input control parameter file the force scaling factor was specified as 5, then constant nodal forces of 5.0 and 10.0, respectively, would be applied along the top of the beam throughout the *NMAP* analysis. At the right end, the beam is subjected to a total displacement of magnitude 0.1, which will be applied in uniform increments during the analysis (i.e., displacement of end nodes will be 0 at the first time-step and will uniformly increase to 0.1 at the final time-step).

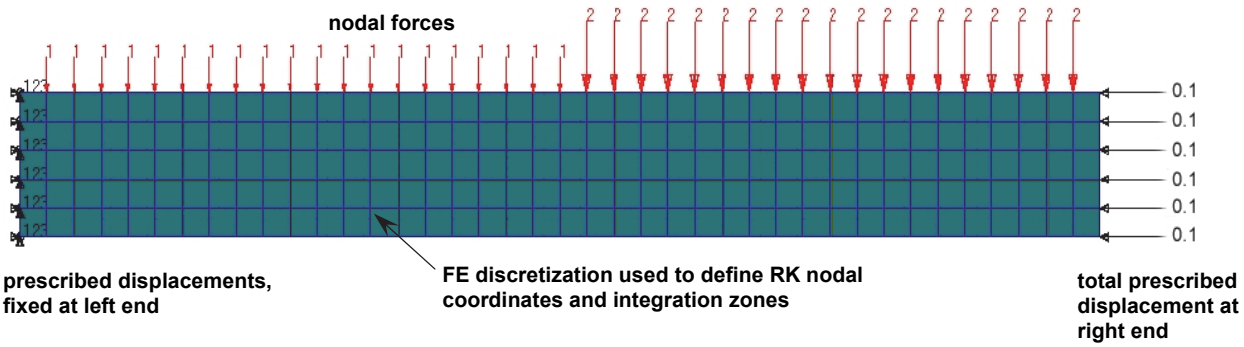


Figure 1. Example of model geometry and boundary conditions defined in *Patran*.

In addition to model geometry and boundary conditions, the user must define material properties, initial conditions, and a variety of control parameters for the *NMAP* analysis. These data are provided to *NMAP_PREPRO* in card format in the input control file *NMAP_Input.dat*. The same file is also used to provide control parameters to the main *NMAP* program and is described in detail in Chapter 3.

Using model data defined in the neutral file and control data from *NMAP_Input.dat*, the preprocessing program is used to construct the input files required for an *NMAP* analysis (in addition to the *NMAP_Input.dat* file). A listing of these files with a brief description of each is given next.

input_bound.dat

This file defines Dirichlet or essential boundary conditions. A single number is given on the first line of the input file, specifying the total number of prescribed non-zero displacements. Data in the remainder of the file are in four-column format, where the first column is the node number and the remaining three columns correspond to the *x*-, *y*-, and *z*-translational degrees of freedom. With the exception of fixed boundary conditions, the node number (either positive or negative) is repeated in each of the three columns. Free degrees of freedom are designated with a positive number, and nodes with prescribed non-zero displacements are designated with a negative number. Fixed degrees of freedom (i.e., prescribed displacement with zero magnitude) are designated with an entry of the number zero. These data are provided for each node in the model.

Following the specification of nodal degrees of freedom (i.e., zero, non-zero or free), the second set of data in the file is used to specify the magnitude and direction of non-zero prescribed displacements. These data are also in four-column format, and the number of rows corresponds to the number of nodes with non-zero prescribed displacements. The first column of this data set again contains the node number, and the remaining three columns contain the *x*-, *y*-, and *z*-displacements. In this section only non-zero entries are used to define the prescribed displacements (i.e., entries with zero magnitude are still treated as free degrees of freedom). Non-zero displacements given in *input_bound.dat* have not been scaled by the displacement scaling factor (reference DCURVE in *NMAP_Input.dat*). The displacement scaling factor is applied within the main *NMAP* program.

input_nforce.dat

This file defines Neumann or natural boundary conditions. The first line specifies the total number of nodes with at least one prescribed natural boundary condition, and the remaining lines define the nodal forces. The nodal force data are in four-column format, where the first column contains the node number and the remaining three columns contain the specified nodal forces. Nodal forces given in *input_nforce.dat* have not been scaled by the force scaling factor (see FCURVE in *NMAP_Input.dat*). The force scaling factor is applied within the main *NMAP* program.

input_coor.dat

This file defines nodal coordinate data. The first line specifies the total number of nodes in the model, and remaining lines define the nodal coordinates. The nodal coordinate data are in four-column format, where the first column contains the node number and the remaining three columns contain the *x*-, *y*-, and *z*-coordinate data for each node. The final line of *input_coor.dat* is used to specify Voronoi cell parameters for SCNI integration. The first number specifies the total number of vertices defining the Voronoi diagram, and the second number specifies the number of vertices used to define a single surface of the Voronoi cell.

input_dila.dat

This file defines the support size for each node. The support size data are in four-column format. The first column contains the node number, and the remaining three columns contain the *x*-, *y*-, and *z*-dimensions of a rectangular nodal support zone. The nodal support zones are calculated based on the specified mesh type (i.e., hexahedral or tetrahedral) and integration technique (i.e., SCNI or SNNI), unless a user-specified constant support size is given in *NMAP_Input.dat*. The support sizes given in *input_dila.dat* have not been scaled by the support size scaling factor (reference scaling factor for DCJP in *NMAP_Input.dat*). The support size scaling factor is applied within the main *NMAP* program.

If kernel functions based on a spherical support are specified (ISPLINE=6-10 in *NMAP_Input.dat*), the radius of the support zone is calculated based on an algorithm that utilizes the *x*-, *y*-, and *z*-nodal support sizes given in *input_dila.dat*.

input_id.dat

This file defines the material set and body set assignments for each node. The material and body set assignment data are in three-column format. The first column is used to specify the node number. The second column defines the material set ID assigned to the node, and the third column defines the body set ID assigned to the node (reference description of file *NMAP_Input.dat* for material set and body set ID definitions).

input_initial.dat

This file defines initial conditions. Two numbers are on the first line of the input file, specifying the total number of nodes with prescribed (1) initial displacements and (2) initial velocities. Following the first line, data for initial displacements are in four-column format. The first column specifies the node number, and the remaining three columns specify the prescribed initial displacements in the *x*-, *y*-, and *z*-directions. If no initial displacements are specified, the second line of the file *input_initial.dat* is left blank. Following the initial displacement data, initial velocities are given. The initial velocity data are in similar four-column format with the node number specified in the first column and initial velocities specified in the remaining three columns. If initial displacement and velocity data are specified, a blank line is entered between the two data sets.

XVOL.dat

This file defines nodal volumes. The data are in single-column format, where each line specifies nodal volume (given in sequential node ordering). The file *XVOL.dat* is generated for SNNI calculations only. Nodal volumes for SCNI calculations are calculated within the main *NMAP* program.

In addition to the preceding seven input files, two additional files are generated when SCNI is specified as the integration technique. These files are *boun.dat* and *data.id*, which contain Voronoi cell data for use in *NMAP*. The file *boun.dat* specifies coordinate data for Voronoi cell vertices, and *data.id* specifies connectivity and surface data for the Voronoi cells. Lastly, the file *input_fem.dat* is generated for use with FEM-RKPM coupling and defines element connectivity from the *Patran* model.

Code execution

Once preprocessing is completed and the necessary input files are generated, the main *NMAP* code can be executed. A user interface has not been developed for the *NMAP* code, so the executable must be called from the command line or through a script file.

Restart

A restart function is provided in the *NMAP* code, where the user may restart an analysis using a set of restart files. In the input control file, users

can specify a time-step frequency for the output of restart files. These files contain state and field variable data and other necessary information (such as time-step, etc.) needed to restart the analysis at the time of restart data output. The standard input files (reference preceding sections) used for the initial analysis are also required. Restart files are written only when an unexpected termination occurs.

Data output

In the input control file, the user may specify the type of data to be output from the *NMAP* analysis (reference *NMAP_input.dat*). Data that may be output include updated nodal coordinates, nodal velocities, damage, nodal stresses (total stress at current time), nodal strains (total strain at current time), and von Mises stress (for AFC models only). These data are output in two output files that are written at calculation step intervals specified by the user. The user may also specify whether data are output in ASCII or binary format (reference LBINARY in *NMAP_Input.dat*). A description of the ASCII format for each output file is given in the following paragraphs.

DISP_xxx.OUT

This file contains updated nodal coordinate data, nodal velocity data, and damage data, written at the *xxx* output interval. The format of the data is given as follows:

- Row 1: Total number of nodes in the model.
- Rows 2 through end: Output data for each node.
  - Column 1: Node number.
  - Columns 2 through 4: Updated nodal coordinates at current time-step. The coordinate data are written in *x*-, *y*-, *z*-coordinate order.
  - Column 5: Shear damage variable computed by the AFC and Drucker-Prager models. The shear damage variable ranges from 0 to 1 (0 = no damage and 1 = complete damage).
  - Column 6: Variable IDID from main *NMAP* program, which indicates whether the calculation was performed using RKPM (IDID=3) or FEM-RKPM coupling (IDID=2).
  - Columns 7 through 9: Updated nodal velocities at current time-step. The velocity data are written in *x*-, *y*-, *z*-velocity order.
  - Column 10: Tension damage variable computed by the AFC multiscale model. The tension damage variable ranges from 0 to 1 (0 = no damage and 1 = complete damage).

SIGEP_xxx.OUT

This file contains total stress and strain data at the current time-step and is written at the *xxx* output interval. The format of the data is given as follows:

- Row 1: Total number of nodes in the model.
- Rows 2 through end: Output data for each node.
  - Column 1: Node number.
  - Columns 2 through 7: Updated nodal stresses at current time-step. The stress data are written in *xx-*, *yy-*, *zz-, xy-, yz-, xz*-stress order.
  - Columns 8 through 13: Updated total nodal strains at current time-step. The strain data are written in *xx-*, *yy-*, *zz-, xy-, yz-, xz*-strain order.
  - Column 14: Updated von Mises stress at current time-step. Von Mises stresses are output only for the AFC models.

In addition to the output files previously described, three additional output files may be selected by the user. These files are *NodalMass.OUT*, *SupportSize.OUT*, and *NMAP_Model.OUT*. The file *NodalMass.OUT* contains the code-calculated nodal masses and is in single column format. Each row contains nodal mass data and is provided in sequential node ordering. The file *SupportSize.OUT* contains code-calculated nodal support sizes, scaled in accordance with the user-specified support size scaling parameter. The support size data are in three-column format, where data in columns 1 through 3 correspond to the *x-*, *y-,* and *z*-dimensions of each node's rectangular support, respectively. In the current version, the code does not output the internally calculated support size radius for kernel functions with spherical support (i.e., ISPLINE=6-10). The file *NMAP_Model.OUT* is generated during model initialization and echoes the input data back to file.

Postprocessor

UCLA has provided a translator for viewing certain output data in ParaView (Squillacote 2008), but a packaged postprocessor has not been developed for use with *NMAP, Version 1.0*. Therefore, postprocessing of the output data must be performed using commercially available software or other user-developed codes.

3 Control Parameter Input File Format

As described in the preceding chapter, a total of eight input files (10 for an SCNI calculation) are required to perform an *NMAP* analysis. The majority of these files are automatically generated by the preprocessor; however, the user must also specify a set of model parameters, material properties, and initial conditions for use in the *NMAP* analysis. These data are contained in the control file *NMAP_Input.dat*, which is written in card format. Format of the *NMAP_Input.dat* file and variable definitions (with guidelines for use, as appropriate) are discussed in this chapter.

Input file format

A sample formatted control parameter input file is shown in Figure 2. Lines beginning with an asterisk (*) are used as control card indicators. These are followed by commented lines (indicated by #) that list the variables to be specified on the following line/lines. Variable definitions for the control parameter inputs are given in the section that follows.

```
* RKPM Parameter
# Basis Func Order[NORDER] KERNEL TYPE [ISPLINE] SUPPORT SIZE(0:use vales below, 1: auto by code)[IDILA] Mesh TYPE(H, T)[CMESH]
        1                   4 1                                    H

# [(DCJP(1:3)] used if IDILA = 0
 0.500000E-01 0.500000E-01 0.500000E-01

* Logical Parameters
# [LSEMI] [LFEMRK] GRAVITY[LGRAV] AUTO SUPPORT[LSUPADJ] RESTART[LNEW] (1:new job, 0:restart) LBINARY ICONTACT
 0 0 0 0 1 0

# FREQUENCY OF UPDATING THE SEARCHBOX [IBOXSW] (used when LSEMI is 1)
 200

* Output Control
# [IOutFreq]   [IDISP][ISIGEP]       [IECHO] [ISUPP] [IMASS] [IFORCE] [IINTEROUT]
 100           1 1 1 1 1

* Node Group Definition
# Num of Node Groups [NUM_GROUP]
       2
# Group ID    Type (1: Order, 2: Specify) Node_Beg OR # OF NODES      Node_End      Node_Inc [NODE_GROUP_ID]
       1       1                           1       4000 1

       2 1 4001 4004 1

* Body Set
# Num of Bodies [NUM_BODY]
       2
# Set ID       Node Group SCALING FACTOR FOR DCJP [...] [FRIC_COEFF]
       1              1 0.75
       2              2 0.75

* Material Set
# Num of Sets [NUM_MAT]
       2
# Set ID      Set Type       Node Group [MAT_ID]
       1       1 1
       2 1 2

# Set Info: Parameters
1 29.0000E+6 0.30000E+00 0.000000E+00 0.00000E+00

2 29.0000E+6 0.30000E+00 0.000000E+00 0.00000E+00 0.000000E+00 0.000000E+00 0.000000E-00 0.73395E-03 0.000100E+00
```

Figure 2. Sample control parameter input file, *NMAP_Input.dat*.

```
## EXAMPLES
## 1 Elasticity 0.100E+10 0.200E+00 0.000E+00 0.000E+00
## 2 Soil Cap 0.168E+08 0.400E+00 0.748E-01 0.000E+00 0.264E+06 0.100E+00 0.500E+02 0.230E+04 0.100E-01
## 3 J2 Plast. 0.580E+11 0.160E+00 0.000E+00 0.000E+00 0.910E+07 0.500E-01 0.100E+02 0.277E+04 0.100E-03
## 4 AFC-MACRO 0.689E+10 0.501E+09 0.476E+09 0.100E-01 0.675E+03 0.516E+09 0.200E+00 0.400E+01 0.247E+07
## 0.551E+08 0.250E-02 0.424E+10 0.619E+10 0.682E+11 0.682E+11 0.250E+00 0.580E-09 0.326E-08
## 0.625E+00 0.177E+22 0.100E+01 0.226E+04 0.100E-03 0.120E+04 1 1 0.689E+04
## 4 AFC-MULTI 0.689E+10 0.501E+09 0.476E+09 0.100E-01 0.675E+03 0.516E+09 0.200E+00 0.400E+01 0.247E+07
## 0.551E+08 0.250E-02 0.424E+10 0.619E+10 0.682E+11 0.682E+11 0.250E+00 0.580E-09 0.326E-08
## 0.625E+00 0.177E+22 0.100E+01 0.226E+04 0.100E-03 0.120E+04 1 1 0.689E+04


* Time Integration
# Time_Begin Time_End Time_Incr [TIMS(1:3)]
 0.000000E+00 5.000E-05 0.1000E-08
# [Gamma] >= 0.5
       0.500000E+00
# fCurve(1)          fCurve(2) [FCURVE(1:2)]
 0.000000E+00 0
# DCurve(1)          DCurve(2) [DCURVE(1:2)]
 0.000000E+00 0
# GRAVITY DIRECTION [G_DIR(1:3)]
 0 0 1

* Initial Displacement
# Num of Initial Displacement Set [Num_IDIS]
       0
# Set ID      Node Group    DISPI(1:3)
 1 2 0.1 0 0

* Initial Velocity
# Num of Initial Velocity Set [Num_Vel]
       1
# Set ID      Node Group    VELI(1:3)
 1             1            1000 0 0

* End
```

Figure 2 (continued). Sample control parameter input file, *NMAP_Input.dat*.

Input file variable definitions

Definitions for the model control parameters defined in *NMAP_Input.dat* are given in the following paragraphs.

NORDER

This variable specifies the order of reproducing conditions for the reproducing kernel approximation. Constant basis should typically be used for penetration problems. 0 = constant basis, 1 = linear basis, 2 = quadratic basis.

ISPLINE

This variable specifies the type of kernel function used in the reproducing kernel approximation.

1 = linear B-spline, rectangular support definition.

2 = quadratic B-spline, rectangular support definition.

3 = cubic B-spline, rectangular support definition.

4 = quartic B-spline, rectangular support definition.

5 = quintic B-spline, rectangular support definition.

6 = linear B-spline, spherical support definition.

7 = quadratic B-spline, spherical support definition.

8 = cubic B-spline, spherical support definition.

9 = quartic B-spline, spherical support definition.

10 = quintic B-spline, spherical support definition.

IDILA

This variable indicates whether uniform nodal support size is used throughout the model (i.e., support size is the same for every node) or whether non-uniform nodal support size is used (i.e., support size is based

on distance-to-neighboring-node calculations). Non-uniform nodal support size should be typically used. If non-uniform nodal support size is used, nodal support sizes are automatically calculated during preprocessing. 0 = uniform nodal support size specified by user; 1 = non-uniform nodal support size calculated by code.

CMESH

This variable indicates whether model geometry was generated using a hexahedral or tetrahedral mesh in the FE preprocessor. H = 8-node hexahedral mesh; T = 4-node tetrahedral mesh.

DCJP

This variable specifies uniform nodal support size (in the *x*-, *y*-, and *z*-directions, respectively) to be applied to all nodes contained in the model. DCJP is used to define nodal support sizes only if IDILA = 0.

LSEMI

This variable indicates whether the analysis is performed using (a) Lagrangian RK approximation with SCNI integration or (b) semi-Lagrangian RK approximation with SNNI integration. The semi-Lagrangian approximation using SNNI integration should be used for problems that exhibit material separation and fragmentation. For the semi-Lagrangian approximation, shape function calculations are updated at every time-step, but neighboring nodes contained in the i^{th} node's support zone are updated only at IBOXSW frequency. Chapter 5 contains additional information on the RK approximations and integration techniques. 0 = Lagrangian approximation using SCNI integration, 1 = semi-Lagrangian approximation using SNNI integration.

LFEMRK

This variable indicates whether finite element/reproducing kernel coupling will be used in the analysis. FEM-RKPM coupling is included in the current code version but is currently being tested. 0 = analysis is performed using RKPM only; 1 = FEM-RKPM coupling is used.

LGRAV

This variable indicates whether gravity effects will be considered in the analysis. If gravity effects are included, the magnitude and direction are specified in G_DIR. 0 = gravity effects are not included in the analysis; 1 = gravity effects are considered.

LSUPADJ

This variable indicates whether nodal support sizes will be automatically adjusted during the calculation. If automatic support size adjustment is used, then at IBOXSW frequency, the support size of each node is checked and adjusted such that uniform support size (in the *x*-, *y*-, and *z*-directions) is provided based on the neighboring node distances. This automatic support size adjustment should not typically be used for penetration/fragmentation problems. 0 = automatic support size adjustment is not used in the analysis; 1 = automatic support size adjustment is used.

LNEW

This variable indicates whether the analysis is being performed as a new calculation or as a restart of a previous calculation. If the analysis is being performed as a restart, the files INTEROUT, DISPTD, DISPDD, VEL, ACC, ACC0, INTERDAMG, INTERSTRN, INTERPSTRS, INTERSTRS, INTERBACK, INTERCRIT, and DEF_GRA are required. 0 = current analysis is a restart; 1 = current analysis is a new analysis.

LBINARY

This variable indicates whether data will be written to the output files in ASCII or binary format. 0 = ASCII format; 1 = binary format.

ICONTACT

This variable indicates the contact algorithm to be used for multi-body contact. 1 = natural kernel contact; 2 = frictional kernel contact with level set contact surface definition; 3 = frictional kernel contact with nodal orientation contact surface definition. Chapter 5 provides information on the contact algorithms.

IBOXSW

This variable indicates the number of time-steps between neighboring node searches for a semi-Lagrangian approximation (not used for Lagrangian approximation). The determination of neighboring nodes is used to define which nodes are contained in the i^{th} node's support zone (which subsequently influences construction of shape functions). Increasing the number of time-steps between performing searches will speed up the calculation, but accuracy may be reduced.

IOUTFREQ

This variable indicates the time-step interval for data output.

IDISP

This variable indicates whether displacement, velocity, and damage data will be output to files *DISP_xxx.OUT*. 0 = data are not output; 1 = data are output.

ISIGEP

This variable indicates whether stress and strain data will be output to files *SIGEP_xxx.OUT*. 0 = data are not output; 1 = data are output.

IECHO

This variable indicates whether input data will be echoed to file *NMAP_Model.OUT*. 0 = data are not echoed; 1 = data are echoed.

ISUPP

This variable indicates whether nodal support data will be output to file *SupportSize.OUT*. 0 = data are not output; 1 = data are output.

IMASS

This variable indicates whether nodal mass data will be written to file *NodalMass.OUT*. 0 = data are not output; 1 = data are output.

IFORCE

This variable is not currently used; set equal to zero.

IINTEROUT

This variable specifies the time-step interval for output of restart files.

NUM_GROUP

This variable indicates the number of node groups defined in the model. The node group definitions are used to designate groups of nodes that will be assigned different material properties, body assignments, and/or initial conditions.

NODE_GROUP_ID

This variable defines nodal assignments to different node groups. The basic format of NODE_GROUP_ID consists of five entries for each node group as follows:

- First column: Node group identification number. This is an identifier assigned to the node group being defined.
- Second column: Node group type. The group type indicates how the node group data will be entered. For group type 1, the beginning and ending node numbers of the group are specified in columns 3 and 4, and the node increment for the group assignment is given in column 5. For group type 2, node numbers are manually entered in free-field format on the lines immediately following. If group type 2 is specified, the third column of data contains the total number of nodes in the group, and no entries are made in columns 4 and 5.
- Third column: For group type 1, the beginning number of the node group is entered. For group type 2, the total number of nodes in the group is entered.
- Fourth column: For group type 1, the ending number of the node group is entered. Column 4 is not used for group type 2.
- Fifth column: For group type 1, node increment for the assignment of nodes to the specified group (i.e., if nodes are assigned to the group sequentially from the first node to the last, enter a value of 1). Column 5 is not used for group type 2.

The total number of node group definitions must equal NUM_GROUP.

NUM_BODY

This variable indicates the number of bodies defined in the model. Bodies are assigned to node groups in BODY_ID and are used to define separate groups of nodes that will interact in accordance with *NMAP*'s contact algorithm for separate bodies.

BODY_ID

This variable is used to assign node groups to different bodies in the model. The basic format of BODY_ID consists of three entries for each body definition. The first column is the body set identification number. This is an identifier assigned to the body being defined. The second column is the node group assigned to the specified body, where the node group must be defined in NODE_GROUP_ID. The third column is a support size scaling factor that is uniformly applied to the support zone of all nodes contained in the body set. The number of data lines entered under variable BODY_ID must be equal to NUM_BODY.

FRIC_COEFF

This variable defines the coefficient of friction to be used in the frictional kernel contact algorithms for multi-body contact. Coefficients of friction are defined for each body set in BODY_ID. When two bodies interact during the analysis, the effective coefficient of friction between them is computed as the average of their coefficients defined in FRIC_COEFF.

NUM_MAT

This variable indicates the number of material sets used in the model. A material set defines the type of material assigned to a node group. Material sets are assigned to node groups in MAT_ID.

MAT_ID

This variable is used to assign material sets to different node groups. The basic format of MAT_ID consists of three entries for each material set definition. The first column of MAT_ID is the material set identification number. This is an identifier assigned to the material set being defined. The second column defines the material set type, which indicates the type of material law being assigned. Currently, five constitutive laws are available in *NMAP* and are as follows:

- Set type 1: Linear elastic material,
- Set type 2: Drucker-Prager geomaterial with damage,
- Set type 3: J2 plasticity with isotropic or kinematic hardening (rate independent),
- Set type 4: AFC model, macroscale formulation, and
- Set type 5: AFC model, multiscale formulation.

The third column of MAT_ID assigns the material set to a specific node group. The node groups must be defined in NODE_GROUP_ID. The number of data lines entered under variable MAT_ID must be equal to NUM_MAT.

Immediately following the MAT_ID data, material property parameters for the specified material data sets must be provided in free-field format. The material property data must be provided in the same order in which material sets are listed in MAT_ID (i.e., the first set of material property data must correspond to set ID 1, etc.). The first column of each material property set is the material set ID, followed by the required material properties. Material property parameters for each of the constitutive laws are given in Tables 1 through 4 at the end of this chapter.

TIMS

This variable defines start time, end time, and time-step for the model. The first column is analysis start time, the second column is end time, and the third column is the analysis time-step.

GAMMA

This variable indicates the Newmark time integration parameter. Second order accuracy in the temporal integration scheme is achieved by using GAMMA = ½. GAMMA < ½ will result in unconditional model instability. GAMMA > ½ will provide algorithmic damping, but model accuracy reduces to first order.

FCURVE

This variable specifies the scaling factor for prescribed nodal forces. Data are provided in two-column format, where the first column defines the variable FCURVE(1) and the second column defines FCURVE(2). The scaling factor is equal to FCURVE(2)-FCURVE(1).

DCURVE

This variable specifies the scaling factor for prescribed non-zero displacements. The data are provided in two-column format where the first column defines the variable DCURVE(1) and the second column defines DCURVE(2). The scaling factor is equal to DCURVE(2)-DCURVE(1).

G_DIR

This variable defines acceleration due to gravity and the coordinate direction in which it acts. Three entries are required, corresponding to the global *x*-, *y*-, and *z*-directions in the model. Acceleration due to gravity is entered in the appropriate column, and zero is entered in the others. In the code, body forces are calculated as $F = -ma$, where a is acceleration due to gravity. Users should account for the (-) sign in the body force calculation when specifying G_DIR.

NUM_IDIS

This variable specifies the number of initial displacement sets. An initial displacement set is used to assign initial nodal displacements to a certain node group. Initial displacements are assigned to node groups in DISPI.

DISPI

This variable assigns initial displacements to nodes contained in a node group. The basic format of DISPI consists of five entries for each displacement set definition. The first column is the displacement set identification number. This is an identifier assigned to the displacement set being defined. The second column is the node group assigned to the displacement set, where the node group must be defined in NODE_GROUP_ID. The third through fifth columns contain the specified displacements in the *x*-, *y*-, and *z*-directions, respectively. The number of data lines entered under variable DISPI must be equal to NUM_IDIS.

NUM_VEL

This variable indicates the number of initial velocity sets. An initial velocity set is used to assign initial nodal velocities to a certain node group. Initial velocities are assigned to node groups in VELI.

VELI

This variable assigns initial velocities to nodes contained in a node group. The basic format of VELI consists of five entries for each velocity set definition. The first column is the velocity set identification number. This is an identifier assigned to the velocity set being defined. The second column is the node group assigned to the velocity set, where the node group must be defined in NODE_GROUP_ID. The third through fifth columns contain the specified velocities in the *x*-, *y*-, and *z*-directions, respectively. The number of data lines entered under variable VELI must be equal to NUM_VEL.

* END

This variable specifies the end of the control parameter input file and must be included as the final line in all *NMAP_Input.dat* files.

Table 1. Constitutive model parameters, linear elastic material (material type 1).

E	Young's modulus
NU	Poisson's ratio
RHO	Density
DAMP	Mass proportional damping

Table 2. Constitutive model parameters, Drucker-Prager geomaterial with damage (material type 2).

E	Young's modulus
NU	Poisson's ratio
B	Yield surface parameter
XH	Yield surface parameter
XK	Yield surface parameter
A	Parameter for damage accumulation function
BB	Parameter for damage accumulation function
RHO	Density
DAMP	Mass proportional damping

Table 3. Constitutive model parameters, J2 rate-independent plasticity with isotropic or kinematic hardening (material type 3).

E	Young's modulus
NU	Poisson's ratio
BETA	Yield surface hardening parameter
H	Yield surface hardening parameter
YIELD	Yield strength
CE1[a]	Yield surface hardening parameter
FTOL	Tolerance for checking yield function
RHO	Density
DAMP	Mass proportional damping
[a] For CE1=1, the bilinear elastic-plastic model is hard coded in NMAP code. For CE1≠1, the material model follows the power law.	

Table 4. Constitutive model parameters, AFC model (macroscale and multiscale).

G	Shear modulus
C1	Failure surface parameter
C2	Failure surface parameter
C3	Failure surface parameter
C4	Failure surface parameter
C5	Failure surface parameter
Q1	Artificial bulk viscosity parameter
Q2	Artificial bulk viscosity parameter
PMIN	Maximum allowable tensile pressure
C6	Equation of state parameter
C7	Equation of state parameter
C	Equation of state parameter
D	Equation of state parameter
S	Equation of state parameter
C9	Equation of state parameter
C10	Equation of state parameter
D1	Damage evolution equation parameter
AN	Failure surface parameter
TXETXCR	Extension failure surface parameter
PRECRIT	Extension failure surface parameter
HMIN	Artificial bulk viscosity parameter
RHO	Density
DAMP	Mass proportional damping
Fc	Tensile strength parameter for use in multiscale model
IDAM	Indicates if damage is to be computed (0 = no, 1 = yes)
IFAIL	Indicates if damage is applied to failure surface (0 = no, 1 = yes)
CONVERT	Multiplication parameter to convert between systems of measurement; CONVERT multiplies all dimensioned variables indicated.

4 Code Structure

NMAP, Version 1.0 is written in the Fortran programming language and is structured with two master routines and 37 subroutines. A flow chart for the overall code structure and flow charts for several primary subroutines are provided in this chapter. A listing of the remaining subroutines with a brief description of each is also provided.

Main code structure

The *NMAP* code is structured around two master routines, *RKPM3D.f* and *smain.f*. *RKPM3D.f* is the entry point into the code and calls the appropriate subroutine to read model data from the input files. After the input data are read, control is passed to the routine *smain.f*, which begins the Newmark time integration algorithm for solution of the dynamic problem. *Smain.f* performs the Newmark predictor calculation, calls the appropriate subroutines to form the internal force vector, and then performs the Newmark corrector calculation for the solution at each time-step. On reaching the user-specified total analysis time, *smain.f* exits back to *RKPM3D.f,* and the analysis is completed. To provide efficiency in variable declarations and data passing, global variables used throughout the code are defined in the module *MOD_NMAP.f*. A flow chart for *NMAP*'s primary code structure is given in Figure 3.

Master routines and primary subroutine flow charts

Basic flow charts for the master routines *RKPM3D.f* and *smain.f* and for four primary subroutines are shown in Figures 4 through 9.

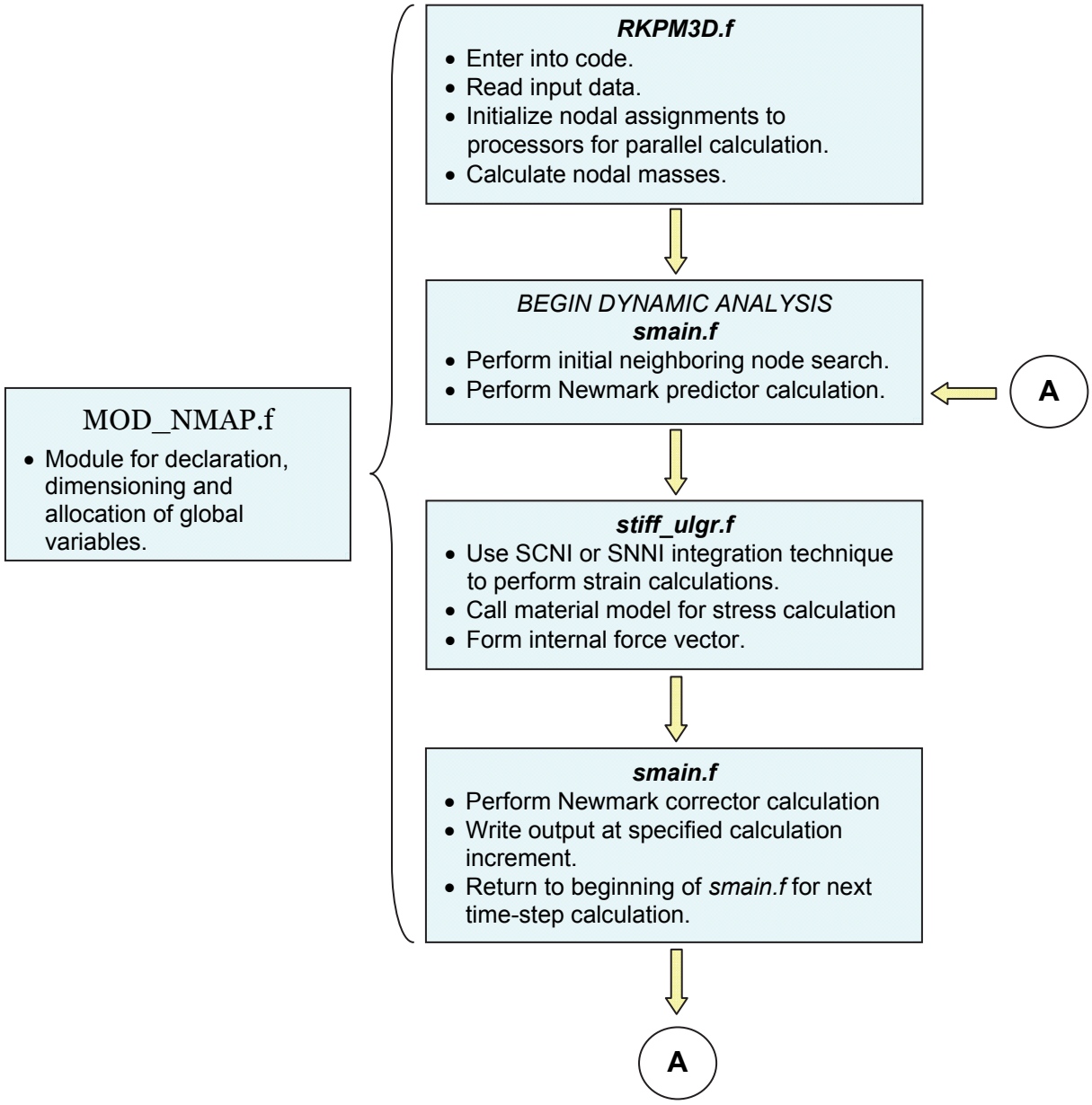


Figure 3. NMAP primary code structure.

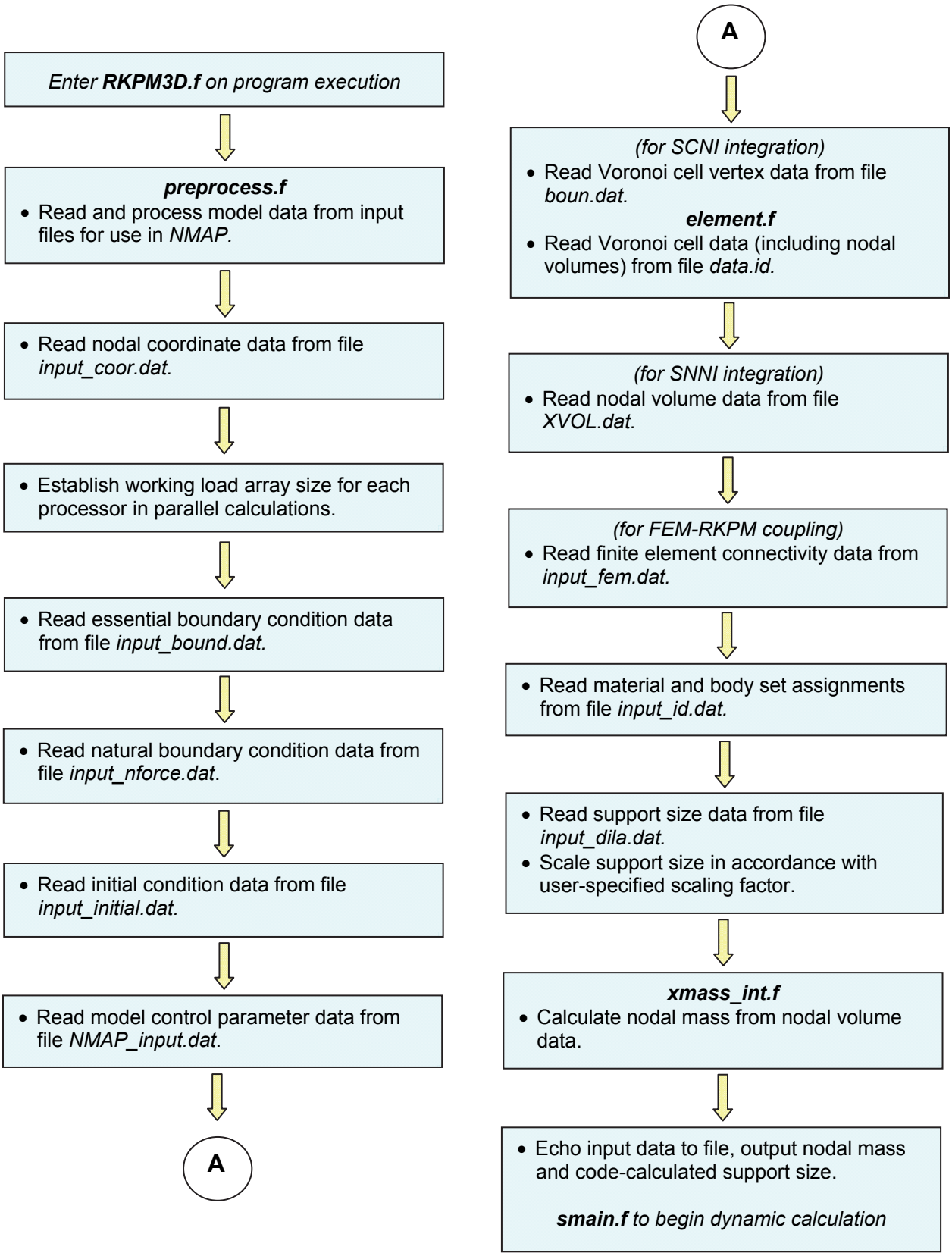


Figure 4. Flow chart, *RKPM3D.f.*

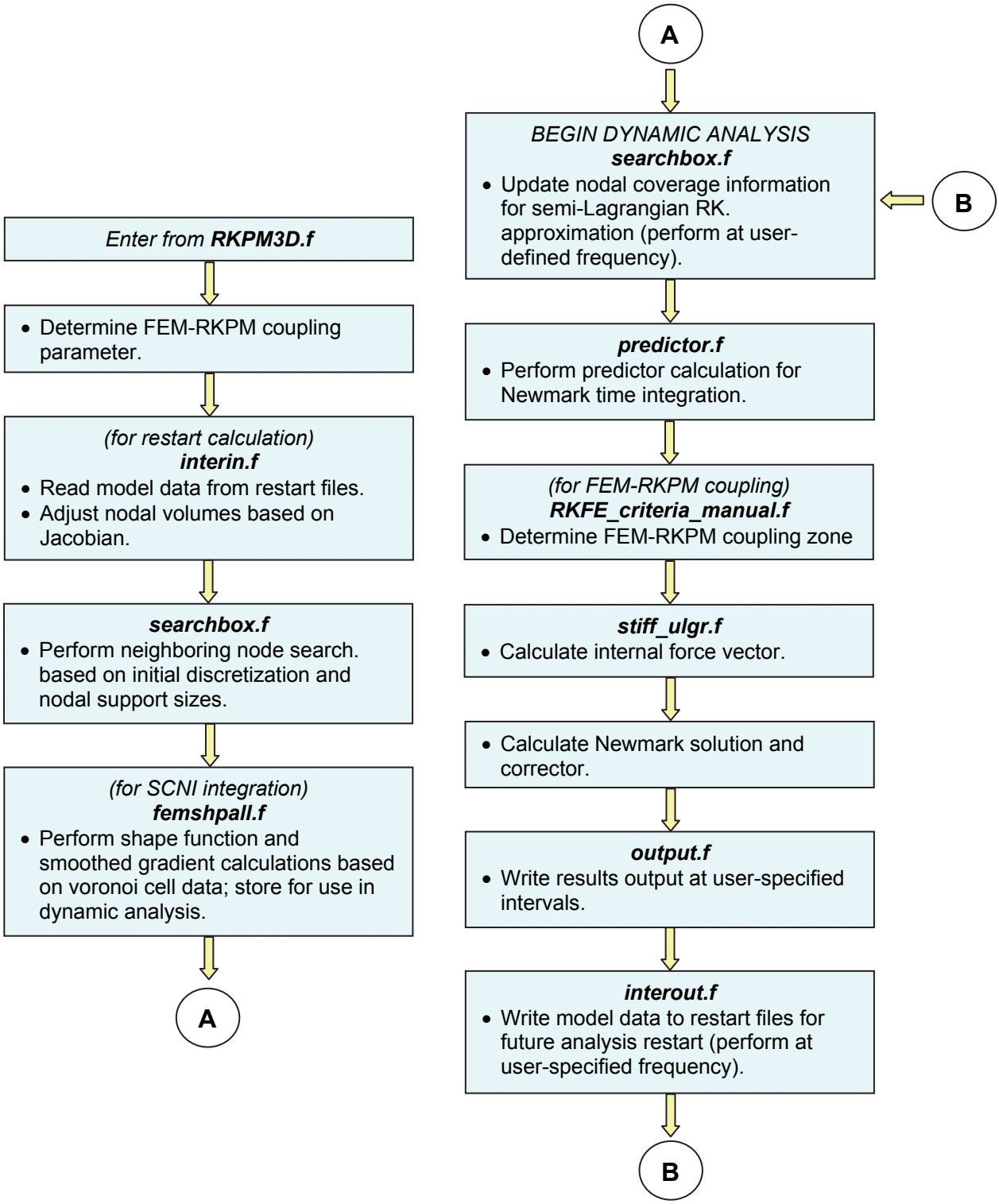


Figure 5. Flow chart, *smain.f.*

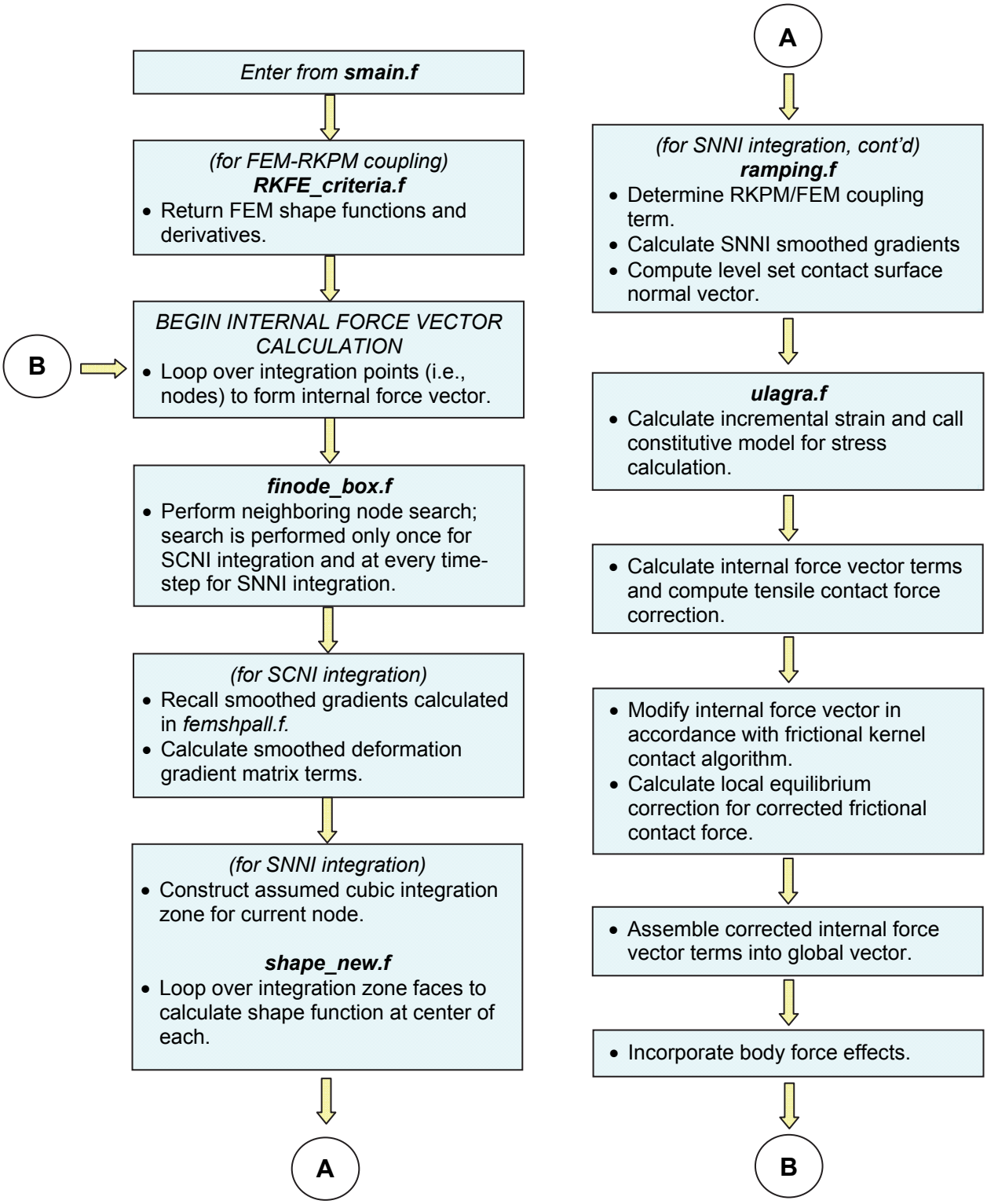


Figure 6. Flow chart, *stiff_ulgr.f.*

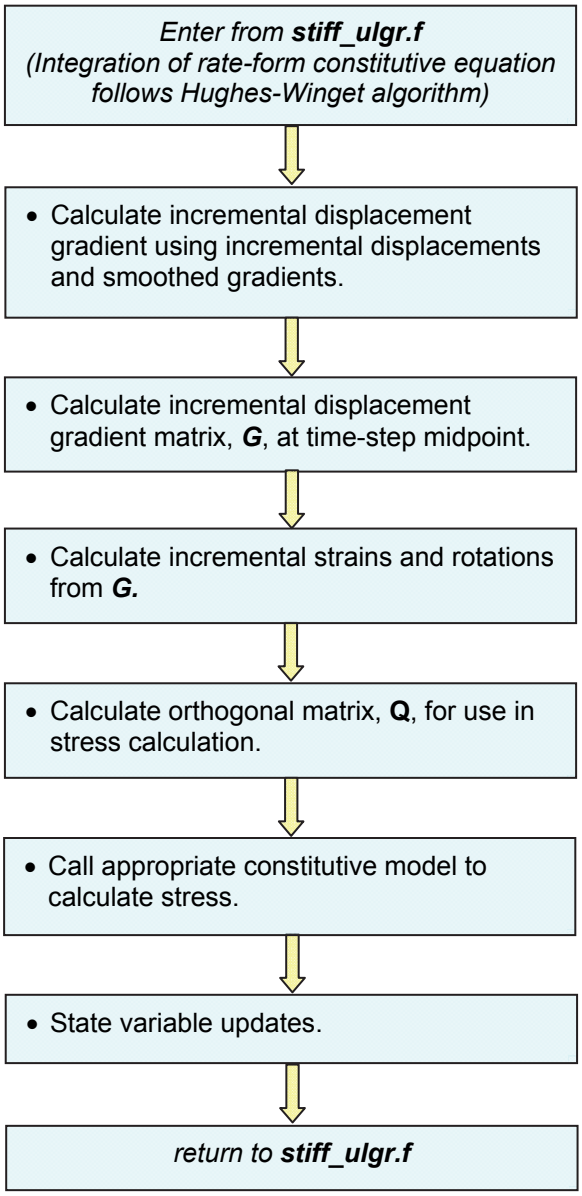


Figure 7. Flow chart, *ulagra.f.*

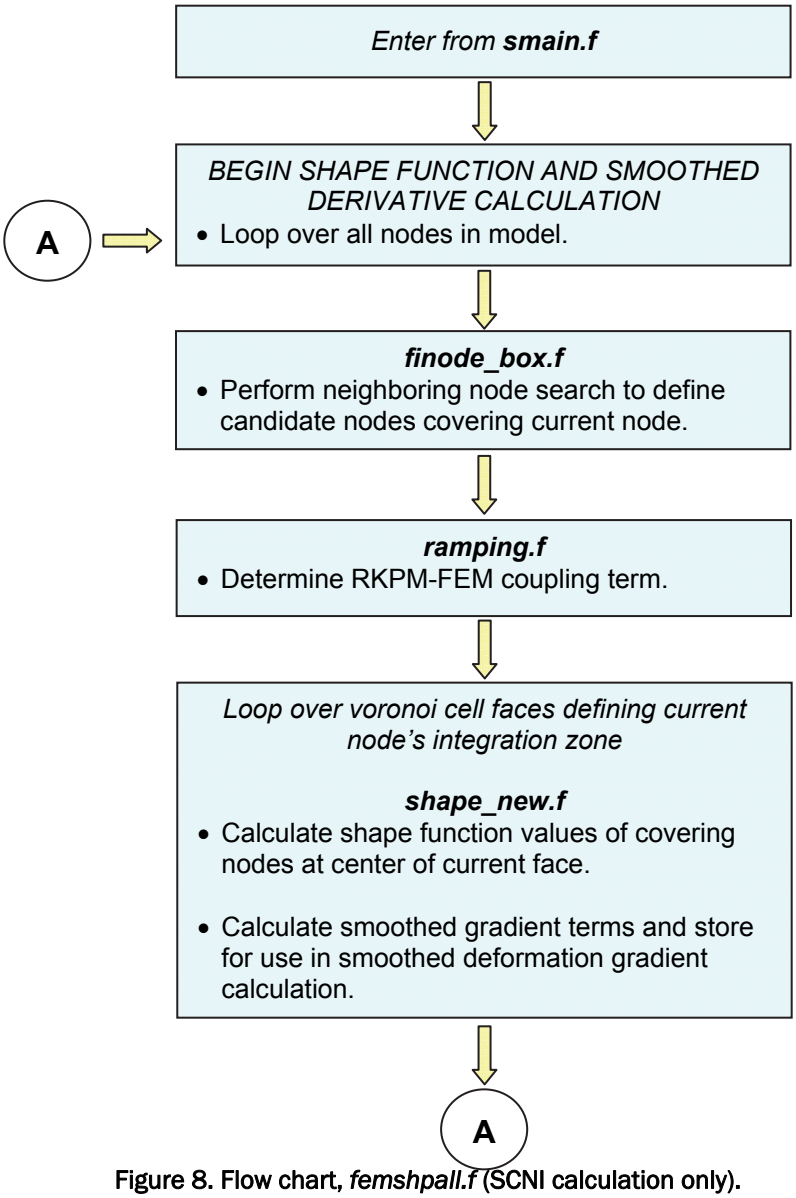


Figure 8. Flow chart, *femshpall.f* (SCNI calculation only).

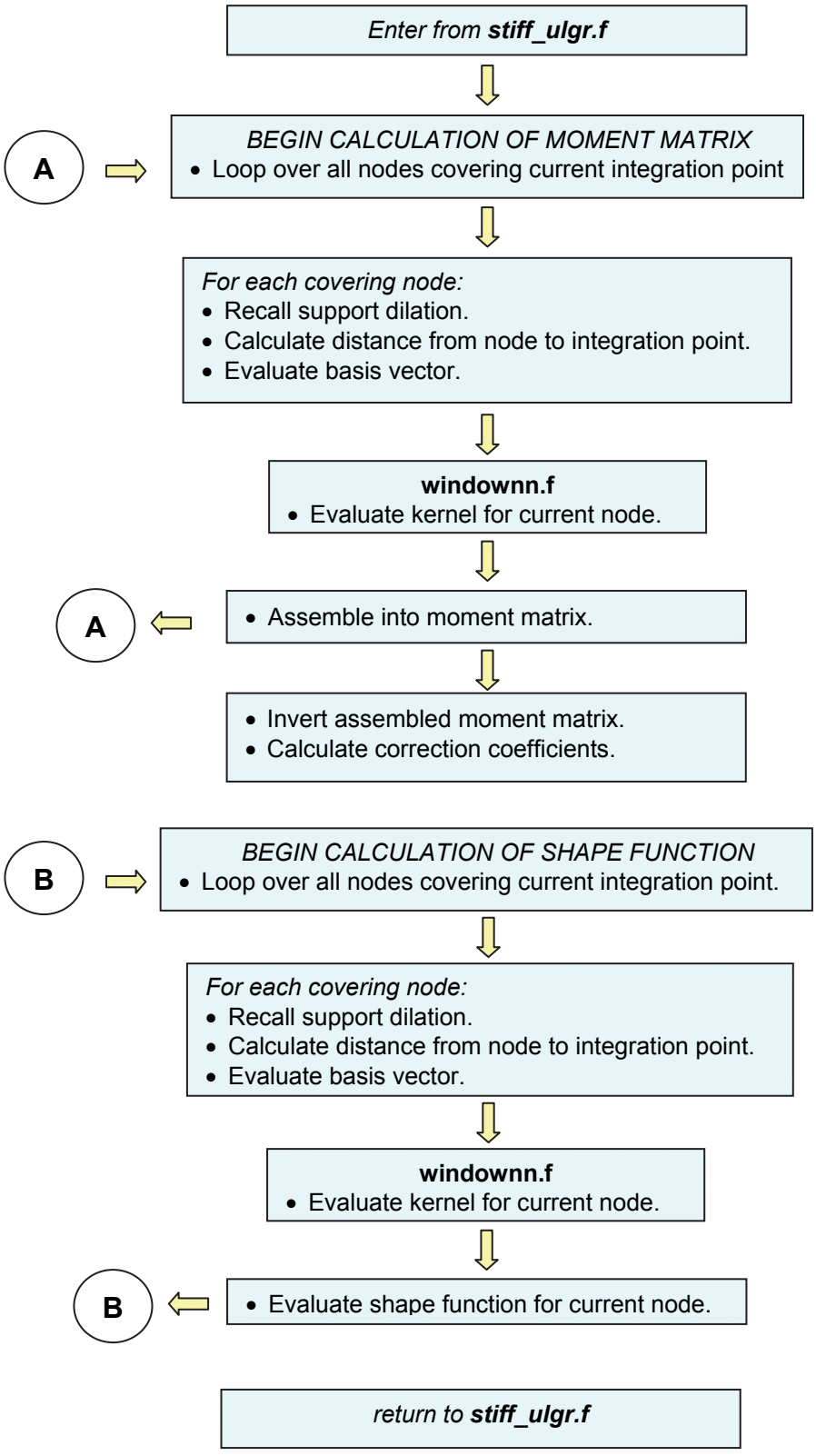


Figure 9. Flow chart, *shape_new.f.*

Secondary subroutines

In addition to the primary routines and subroutines described in the preceding section, 33 additional subroutines are included in the *NMAP, Version 1.0* code package. These subroutines are listed by functional categories in the following paragraphs with a brief description given for each.

Module subroutine

MOD_NMAP.f: Global variable declarations.

Input data subroutines

Sub_preprocess.f: Allocate and initialize variables; read and process model input data; make nodal assignments to processors for parallel calculation.

Sub_xmass.f: Calculate nodal masses from nodal volume data.

Sub_element.f: Read in Voronoi cell data (only for SCNI integration).

Shape function and support zone calculation subroutines

Sub_search_box.f: Calculate coverage for i^{th} node based on enlarged support size. Nodes are identified as candidate covering nodes, which are refined in *sub_finode_box.f*.

Sub_finode_box.f: Calculate nodal coverage for i^{th} node based on enlarged support size. The routine is similar to *sub_search_box.f*, but with a smaller search box. Nodes are identified as candidate covering nodes, which are later refined based on exact support size.

Sub_upd_support.f: Automatically adjust support sizes if an automatic support size update is specified in the input file.

Sub_window.f: Calculate the window or kernel function value at a specified point for use in construction of the reproducing kernel (RK) shape function.

Sub_kernel.f: Calculate a one-dimensional (1-D) kernel function value at a specified point for use in constructing a window function in *sub_window.f*.

Temporal integration subroutines

Sub_predictor.f: Perform the Newmark predictor phase calculation and calculate prescribed incremental displacements from user-specified total displacements.

Material model subroutines

AFC_model_macro.f: Macroscale AFC model.

AFC_model_multi.f: Multiscale AFC model.

AFC_function.f: Function to include separate extension failure surface in AFC model calculation (based on third-invariant of deviatoric stress).

Sub_pmod01.f: Drucker-Prager geomaterial model.

Sub_pmod05.f: Rate-independent plasticity model with kinematic or isotropic hardening.

Sub_EM.f: Linear elastic material model.

Output subroutines:

Sub_output.f: Output user-specified model results to output files.

FEM-RKPM coupling subroutines

Sub_rkfe_criteria: Return FEM shape function and derivative.

Sub_rkfe_criteria_manual: Specify RKPM-FEM coupling zone for circular region.

Sub_femshaped: Calculate FEM shape function derivative.

Sub_femshp_inv_3D_Newton: Calculate FEM shape function.

Sub_findelem: Determine elements connected to the i^{th} node.

Sub_ramping: Calculate FEM-RKPM coupling term.

Miscellaneous subroutines

Sub_interin: Read model data for restart calculation.

Sub_interout: Write model data for restart calculation.

Sub_mics: Set two vectors equal; create null vector.

Sub_blaslib.f: Contains LAPACK driver routine.

Sub_lapack.f: Contains LAPACK routine.

Sub_inver.f: Calculate inverse of a matrix.

Sub_qqq.f: Output character strings to file for error handling.

Sub_condnumber.f: Calculate condition number.

Sub_tnorm: Compute the norm of a tensor.

Sub_new_check_dtcr.f: Calculate the critical time-step and dynamically adjust.

5 Theory

Reproducing kernel approximation

This chapter provides a review of the RK approximation, which is the foundation of RKPM. Consider a domain $\bar{\Omega}$ discretized by a set of nodes $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{NP}\}$, $\boldsymbol{x}_I \in \bar{\Omega}$, $I = 1, 2, \ldots, NP$ and *NP* is the number of points. The RK approximation of a function *u*, denoted by u^h, is expressed as (Liu, Jun, and Zhang 1995; Chen et al. 1996)

$$u^h(\boldsymbol{x}) = \sum_{I=1}^{NP} \Psi_I(\boldsymbol{x}) d_I \tag{1}$$

where $\Psi_I(\boldsymbol{x})$ is the RK shape function and d_I is the corresponding coefficient. The RK shape function is constructed with the following form:

$$\Psi_I(\boldsymbol{x}) = C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I)\varphi_a(\boldsymbol{x} - \boldsymbol{x}_I) \tag{2}$$

where $\boldsymbol{x}_I$ is the nodal position vector, $\varphi_a(\boldsymbol{x} - \boldsymbol{x}_I)$ is the kernel function, and $C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I)$ is the correction function.

The kernel function $\varphi_a(\boldsymbol{x} - \boldsymbol{x}_I)$ is a compactly supported positive function

$$\begin{cases} \varphi_a(\boldsymbol{x} - \boldsymbol{x}_I) \geq 0, & \|\boldsymbol{x} - \boldsymbol{x}_I\| / a \leq 1 \\ \varphi_a(\boldsymbol{x} - \boldsymbol{x}_I) = 0, & \|\boldsymbol{x} - \boldsymbol{x}_I\| / a > 1 \end{cases} \tag{3}$$

where *a* is the measure of support of $\varphi_a(\boldsymbol{x} - \boldsymbol{x}_I)$. The kernel function expressed in Equation 3 has a spherical support with radius *a*. A kernel function with a rectangular or cubic support can be constructed by multiplication of 1-D kernel functions (Chen et al. 1996). The correction function $C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I)$ is the combination of complete n^{th} order monomials

$$\begin{aligned} C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I) &= \sum_{i+j+k=0}^{n} b_{ijk}(\boldsymbol{x})(x_1 - x_{I1})^i (x_2 - x_{I2})^j (x_3 - x_{I3})^k \\ &= \boldsymbol{H}^T(\boldsymbol{x} - \boldsymbol{x}_I)\boldsymbol{b}(\boldsymbol{x}) \end{aligned} \tag{4}$$

$$\boldsymbol{H}^T(\boldsymbol{x}-\boldsymbol{x}_I)=\begin{bmatrix}1 & x_1-x_{I1} & x_2-x_{I2} & x_3-x_{I3} & (x_1-x_{I1})^2 & \dots & (x_3-x_{I3})^n\end{bmatrix} \quad (5)$$

where $b_{ijk}(\boldsymbol{x})$ are the coefficients of the basis functions and $\boldsymbol{b}(\boldsymbol{x})$ and $\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I)$ are vectors of the coefficients and monomial basis functions, respectively. The coefficient vector $\boldsymbol{b}(\boldsymbol{x})$ is solved by enforcing the exact reproduction of the monomial bases up to the n^{th} order.

$$\sum_{I=1}^{NP}\Psi_I(\boldsymbol{x})x_{I1}^i x_{I2}^j x_{I3}^k = x_1^i x_2^j x_3^k \qquad i+j+k=0,1,\dots,n \quad (6)$$

Equation 6 can be transformed to the following.

$$\sum_{I=1}^{NP}\Psi_I(\boldsymbol{x})(x_1-x_{1I})^i(x_2-x_{2I})^j(x_3-x_{3I})^k=\delta_{i0}\delta_{j0}\delta_{k0} \qquad i+j+k=0,1,\dots,n \quad (7)$$

By substituting Equations 2 and 4 into Equation 7, the coefficient vector $\boldsymbol{b}(\boldsymbol{x})$ is obtained as:

$$\boldsymbol{b}(\boldsymbol{x})=\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{H}(\boldsymbol{0}) \quad (8)$$

where

$$\begin{aligned}\boldsymbol{M}(\boldsymbol{x})&=\sum_{J=1}^{NP}\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_J)\boldsymbol{H}^T(\boldsymbol{x}-\boldsymbol{x}_J)\varphi_a(\boldsymbol{x}-\boldsymbol{x}_J)\\ \boldsymbol{H}^T(\boldsymbol{0})&=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & \dots & 0\end{bmatrix}\end{aligned} \quad (9)$$

.oFinally, the RK shape function is obtained as

$$\Psi_I(\boldsymbol{x})=\boldsymbol{H}^T(\boldsymbol{0})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I)\varphi_a(\boldsymbol{x}-\boldsymbol{x}_I) \quad (10)$$

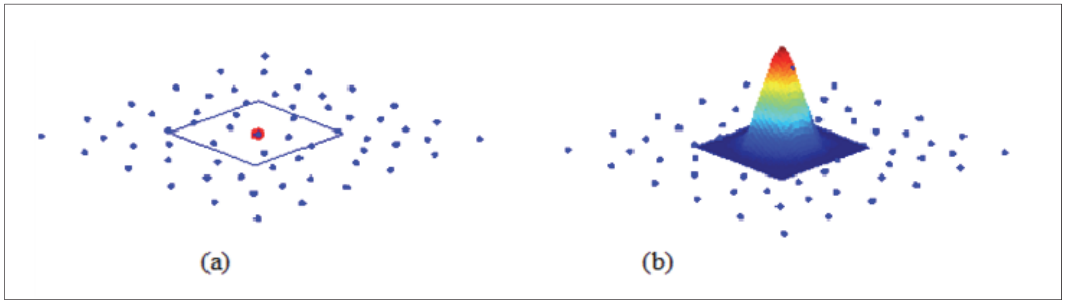


Figure 10 shows the contour plot of a 2-D RK shape function with rectangular support. The kernel support is shown on the left, thus defining the domain of influence; and the shape function is shown on the right, constructed using a cubic B-spline kernel function and linear bases.

Figure 11 shows the comparison of RKPM discretization with circular support and an FEM triangular mesh using the same set of points. The domain of influence of each FEM node is determined by the neighboring connected elements, whereas the domain of influence of the RK shape function is defined by the support of the kernel function. While in RKPM discretization some domains of influence are extended outside of the physical boundary, the reproducing conditions enforced in Equation 6 guarantee the order of accuracy for all $\boldsymbol{x} \in \bar{\Omega}$. This extended boundary layer in RKPM needs to be considered in contact problems. However, it serves as an "insulation layer" to ensure impenetration conditions in the normal contact similar to the function of a "gap element" in the finite element setting.

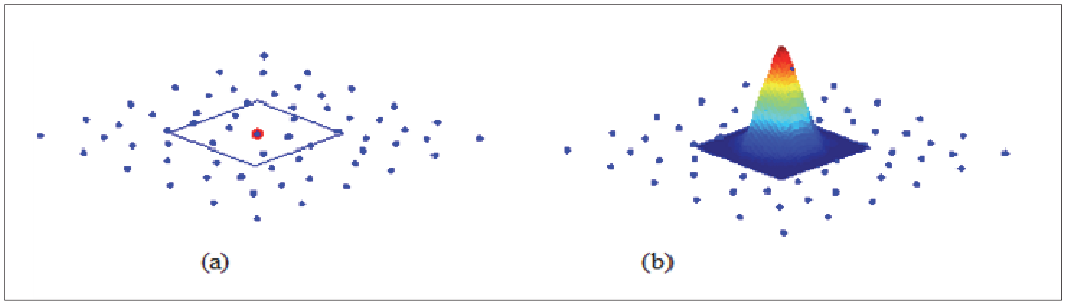


Figure 10. Contour for 2-D RKPM shape function with rectangular support: (a) domain of influence and (b) RK shape function.

Lagrangian/semi-Lagrangian formulation

Updated Lagrangian equation of motion

For modeling of fragment-impact processes, a semi-Lagrangian RK discretization was introduced to the equation of motion. An updated Lagrangian formulation in which the current configuration was the referenced configuration was the beginning point, and a semi-Lagrangian RK approximation constructed in the current configuration was introduced to the updated Lagrangian variational equation X is the material coordinate representing the initial position of a material point and x is the current position of the material point X in the current

configuration with domain Ω_x, essential boundary $\partial\Omega_x^g$, and natural boundary $\partial\Omega_x^h$. The weak form of the equation of motion is:

$$\int_{\Omega_x} \delta u_i \rho \ddot{u}_i d\Omega + \int_{\Omega_x} \delta u_{(i,j)} \tau_{ij} d\Omega = \int_{\Omega_x} \delta u_i b_i d\Omega + \int_{\partial\Omega_x^h} \delta u_i h_i d\Gamma \tag{11}$$

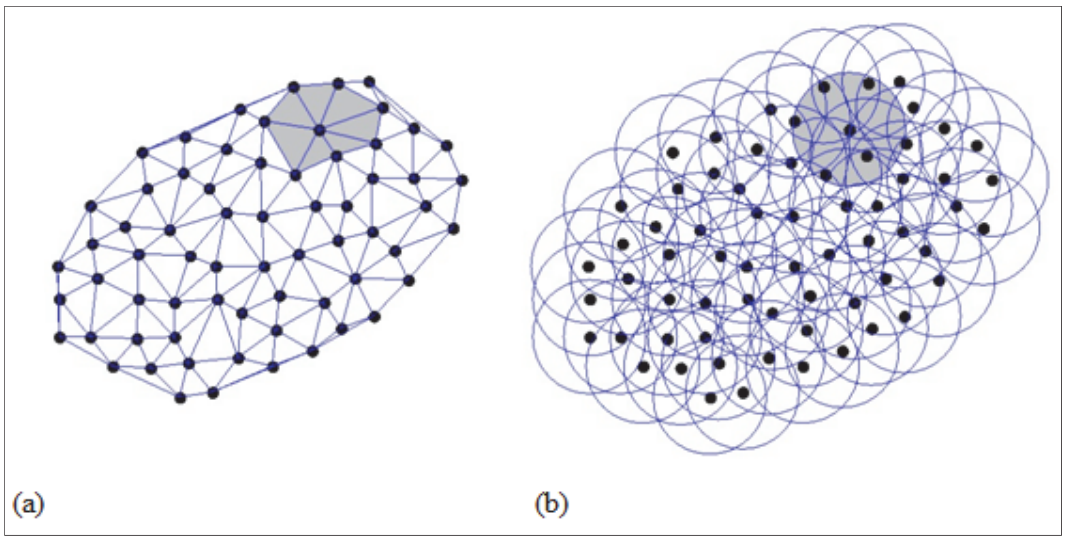


Figure 11. Comparison of FEM and RKPM discretizations and domains of influence; (a) FEM discretization and (b) RKPM discretization. The domain of influence of one node is marked in grey as an example.

where u_i is the displacement, τ_{ij} is the Cauchy stress, $u_{(i,j)} = (\partial u_i / \partial x_j + \partial u_j / \partial x_i)/2$, and ρ, b_i and h_i are density, body force, and surface traction in the current configuration, respectively. In the pure Lagrangian RKPM formulation, the Lagrangian RK shape functions, $\Psi_I(\boldsymbol{X})$, are constructed using the material coordinates in the initial configuration. The discretization of Equation 11 by the Lagrangian RK approximation requires taking the spatial derivatives of the Lagrangian RK shape function, $\Psi_I(\boldsymbol{X})$, which is obtained by the chain rule as

$$\frac{\partial \Psi_I(\boldsymbol{X})}{\partial x_i} = \frac{\partial \Psi_I(\boldsymbol{X})}{\partial X_j}\frac{\partial X_j}{\partial x_i} = \frac{\partial \Psi_I(\boldsymbol{X})}{\partial X_j} F^{-1}_{ji} \tag{12}$$

Here, the deformation gradient $\boldsymbol{F}$ is first computed using the material spatial derivatives of the Lagrangian RK shape functions, and $\boldsymbol{F}^{-1}$ is then obtained by taking the inversion of $\boldsymbol{F}$ (instead of computing $\boldsymbol{F}^{-1}$ directly).

However, the Lagrangian formulation breaks down when the inverse of $\boldsymbol{F}$ is no longer available. This may occur, for example, when large deformation leads to a non-positive definite $\boldsymbol{F}$ or material separation takes place. Thus, in this project, the semi-Lagrangian RKPM formulation (Guan et al. 2009) was introduced for fragment-impact problems.

The Lagrangian formulation breaks down when mapping between the initial and current configurations is no longer one-to-one. This occurs under conditions such as new free surface formation (i.e., material separation) or free surface closure, which commonly exist in penetration processes. Chen and Wu (2007) proposed a semi-Lagrangian formulation to overcome the shortcoming of the Lagrangian formulation.

Semi-Lagrangian formulation

In the semi-Lagrangian formulation, the RKPM points follow the material motion, while the distance measure $z = \boldsymbol{x} - \boldsymbol{x}(\boldsymbol{X}_I, t)$ in the kernel function $z = \boldsymbol{x} - \boldsymbol{x}(\boldsymbol{X}_I, t)$ is defined in the deformed configuration. Under this construction, the kernel support of the semi-Lagrangian kernel function does not deform with the material motion. A comparison of the supports of Lagrangian and semi-Lagrangian kernels at undeformed and deformed states is shown in

Figure 12. The Lagrangian kernel supports cover the same group of material nodes before and after deformation, while the semi-Lagrangian kernel supports cover a different group of nodes after deformation.

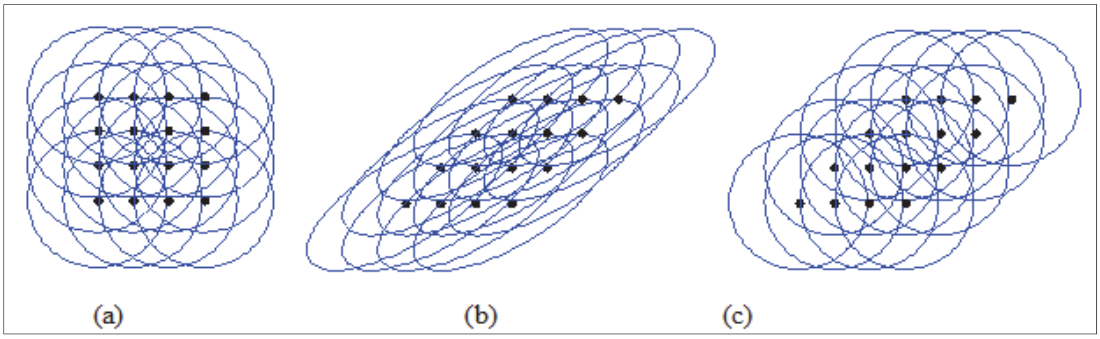


Figure 12. Comparison of Lagrangian and semi-Lagrangian kernel supports in undeformed and deformed configurations: (a) undeformed configuration, (b) Lagrangian kernel deformed with the material in the deformed configuration, and (c) semi-Lagrangian kernel in the deformed configuration.

The semi-Lagrangian RK shape function formulated in the current configuration is expressed as

$$\Psi_I(\boldsymbol{x}) = \boldsymbol{H}^T\left(\boldsymbol{x}-\boldsymbol{x}\left(\boldsymbol{X}_I,t\right)\right)\boldsymbol{b}(\boldsymbol{x})\varphi_a\left(\boldsymbol{x}-\boldsymbol{x}\left(\boldsymbol{X}_I,t\right)\right) \tag{13}$$

The coefficient vector $\boldsymbol{b}(\boldsymbol{x})$ is solved by imposing the semi-Lagrangian reproducing conditions

$$\sum_{I=1}^{NP}\Psi_I(\boldsymbol{x})x_1\left(\boldsymbol{X}_I,t\right)^i x_2\left(\boldsymbol{X}_I,t\right)^j x_3\left(\boldsymbol{X}_I,t\right)^k = x_1^i x_2^j x_3^k, \quad i+j+k=0,1,\ldots,n \tag{14}$$

Solving $\boldsymbol{b}(\boldsymbol{x})$ from Equation 14 and substituting it into Equation 13 yields the following semi-Lagrangian RK shape function:

$$\Psi_I(\boldsymbol{x}) = C\left(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}\left(\boldsymbol{X}_I,t\right)\right)\varphi_a\left(\boldsymbol{x}-\boldsymbol{x}\left(\boldsymbol{X}_I,t\right)\right) \tag{15}$$

where

$$C\left(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}\left(\boldsymbol{X}_I,t\right)\right) = \boldsymbol{H}^T(\boldsymbol{0})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{H}\left(\boldsymbol{x}-\boldsymbol{x}\left(\boldsymbol{X}_I,t\right)\right) \tag{16}$$

$$\boldsymbol{M}(\boldsymbol{x}) = \sum_{I=1}^{NP}\boldsymbol{H}\left(\boldsymbol{x}-\boldsymbol{x}\left(\boldsymbol{X}_I,t\right)\right)\boldsymbol{H}^T\left(\boldsymbol{x}-\boldsymbol{x}\left(\boldsymbol{X}_I,t\right)\right)\varphi_a\left(\boldsymbol{x}-\boldsymbol{x}\left(\boldsymbol{X}_I,t\right)\right) \tag{17}$$

$$\boldsymbol{H}^T\left(\boldsymbol{x}-\boldsymbol{x}\left(\boldsymbol{X}_I,t\right)\right) = \left[1 \quad x_1-x_1\left(\boldsymbol{X}_I,t\right) \quad x_2-x_2\left(\boldsymbol{X}_I,t\right) \quad \cdots \quad \left(x_3-x_3\left(\boldsymbol{X}_I,t\right)\right)^n\right] \tag{18}$$

Let the velocity v_i be approximated by semi-Lagrangian RK shape functions.

$$v_i^h(\boldsymbol{x},t) = \sum_{I=1}^{NP}\Psi_I(\boldsymbol{x})v_{iI}(t) \tag{19}$$

The corresponding semi-Lagrangian approximation of acceleration is given as

$$\ddot{u}_i^h(\boldsymbol{x},t) = \dot{v}_i^h(\boldsymbol{x},t) = \sum_{I=1}^{NP}\left(\Psi_I(\boldsymbol{x})\dot{v}_{iI}(t) + \Psi^*_I(\boldsymbol{x})v_{iI}(t)\right) \tag{20}$$

where $\Psi_I^*(\mathbf{x})$ is the correction due to the time rate of the semi-Lagrangian kernel $\dot{\varphi}_a$

$$\Psi_I^*(\boldsymbol{x}) = C\left(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}\left(\boldsymbol{X}_I, t\right)\right)\dot{\varphi}_a\left(\boldsymbol{x} - \boldsymbol{x}\left(\boldsymbol{X}_I, t\right)\right) \tag{21}$$

$$\dot{\varphi}_a\left(\frac{\left\|\boldsymbol{x} - \boldsymbol{x}(\boldsymbol{X}_I, t)\right\|}{a}\right) = \varphi_a'\left(\frac{\left\|\boldsymbol{x} - \boldsymbol{x}(\boldsymbol{X}_I, t)\right\|}{a}\right)\frac{\boldsymbol{n}\cdot\left(\boldsymbol{v} - \boldsymbol{v}_I\right)}{a} \tag{22}$$

where $\dot{(\ )} = \left.\frac{\partial(\ )}{\partial t}\right|_{[X]}$ is the material time derivative, and

$$\boldsymbol{n} = \frac{\boldsymbol{x} - \boldsymbol{x}\left(\boldsymbol{X}_I, t\right)}{\left\|\boldsymbol{x} - \boldsymbol{x}\left(\boldsymbol{X}_I, t\right)\right\|}. \tag{23}$$

Substituting Equation 20 into Equation 11 yields the following semi-Lagrangian discrete equation:

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{N}\boldsymbol{v} = \boldsymbol{f}^{ext} - \boldsymbol{f}^{\text{int}} \tag{24}$$

where

$$\boldsymbol{M}_{IJ} = \int_{\Omega_x} \rho\Psi_I\left(\boldsymbol{x}\right)\Psi_J\left(\boldsymbol{x}\right)\boldsymbol{I}d\Omega \tag{25}$$

$$\boldsymbol{N}_{IJ} = \int_{\Omega_x} \rho\Psi_I\left(\boldsymbol{x}\right)\Psi_J^*\left(\boldsymbol{x}\right)\mathbf{I}d\Omega \tag{26}$$

$$\boldsymbol{f}_I^{\text{int}} = \int_{\Omega_x} \boldsymbol{B}_I^T\boldsymbol{\Xi}d\Omega \tag{27}$$

$$\boldsymbol{f}_I^{ext} = \int_{\Omega_x} \Psi_I^T\boldsymbol{b}d\Omega + \int_{\partial\Omega_x} \Psi_I^T\boldsymbol{h}d\Gamma \tag{28}$$

where $\boldsymbol{B}_I$ is the gradient matrix associated with $u_{(i,j)}$, $\boldsymbol{\Xi}$ is the stress vector associated with τ_{ij}, and $\boldsymbol{b}$ and $\boldsymbol{h}$ are body force and surface traction vectors, respectively.

SCNI/SNNI integration

Domain integration in Galerkin meshfree methods requires special attention, as there is no mesh in the discretization. Gauss integration requires a background grid and introduces significant integration errors if the kernel supports do not match with the integration grid. Nodal integration with stabilization, such as the stabilized conformation nodal integration (SCNI) (Chen et al. 2001; Chen, Yoon, and Wu 2002), was proposed as an alternative to the Gauss integration. In SCNI, nodal strain smoothing on a conforming smoothing domain surrounding each node is introduced to achieve stability and optimal convergence. The strain smoothing at the node L is calculated by

$$\bar{\varepsilon}_{ij}\left(\boldsymbol{x}_L\right)=\frac{1}{A_L}\int_{\Omega_L}\varepsilon_{ij}d\Omega=\frac{1}{2A_L}\int_{\Omega_L}\left(u_{i,j}+u_{j,i}\right)d\Omega \tag{29}$$

where ε_{ij} and $\bar{\varepsilon}_{ij}$ are components of the regular strain and smoothed strain tensors, respectively, u_i is a component of the displacement vector, and A_L is the area (or volume) of the conforming smoothing domain associated with node *L*. By introducing the divergence theorem, Equation 29 is transformed into a boundary integral to yield

$$\begin{aligned}\frac{1}{2A_L}\int_{\Omega_L}\left(u_{i,j}+u_{j,i}\right)d\Omega&=\frac{1}{2A_L}\int_{\partial\Omega_L}\left(u_i n_j+u_j n_i\right)d\Gamma\\&=\frac{1}{2A_L}\int_{\partial\Omega_L}\sum_{I=1}^{NP}\Psi_I\left(\boldsymbol{x}\right)u_{Ii}n_j+\sum_{I=1}^{NP}\Psi_I\left(\boldsymbol{x}\right)u_{Ij}n_i d\Gamma\\&=\sum_{I=1}^{NP}\left(\bar{b}_{Ij}u_{Ii}+\bar{b}_{Ii}u_{Ij}\right)\end{aligned} \tag{30}$$

where

$$\bar{b}_{Ij}=\frac{1}{A_L}\int_{\partial\Omega_L}\Psi_I\left(\boldsymbol{x}\right)n_j d\Gamma \tag{31}$$

In Equation 31, $\bar{b}_{Ij}$ is the smoothed gradient of the shape function. Strain smoothing on a conforming smoothing domain, that is $\cup_L\bar{\Omega}_L=\bar{\Omega}$, is the requirement to satisfy the integration constraint for optimal convergence

(Chen et al. 2001). One choice for generation of the conforming smoothing domains is the Voronoi diagram, as shown in

Figure 13. However, SCNI comes with the cost of constructing the conforming smoothing domains, and it is particularly tedious in the semi-Lagrangian discretization where smoothing domain reconstruction at every time-step is needed. For penetration problems, the damage evolution and the associated new surface formation further complicate the Voronoi cell generation. In this work, the stabilized non-conforming nodal integration (SNNI) (Chen et al. 2006; Chen and Wu 2007) is introduced. In this approach, the conforming requirement in the nodal strain smoothing domain is not enforced, that is, $\cup_L \bar{\Omega}_L \neq \bar{\Omega}$ as shown in

Figure 14, where simple strain-smoothing domains are used. RKPM with SNNI remains stable, with accuracy comparable to that with SCNI (Chen et al. 2006; Chen and Wu 2007).

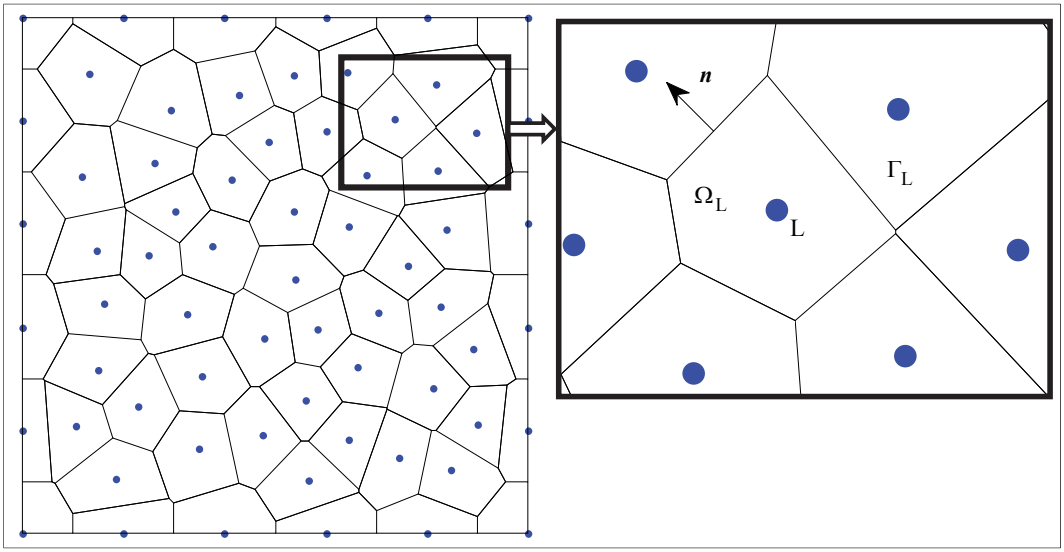
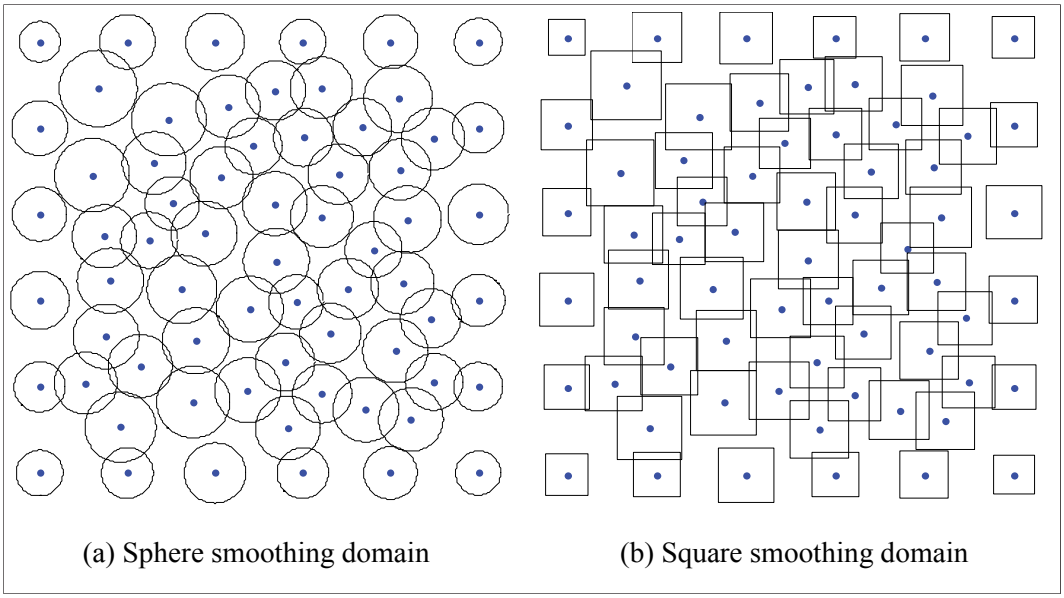


Figure 13. Strain-smoothing domains for SCNI.



(a) Sphere smoothing domain (b) Square smoothing domain

Figure 14. Strain smoothing domains for SNNI.

Enhanced kernel contact algorithm

Kernel contact algorithm

Conventional contact algorithms, such as the penalty method, require the potential contact surfaces to be predefined. However, for penetration problems, contact surfaces are changing continuously, and they cannot be defined a priori. Here, a kernel contact algorithm that utilizes the interaction of kernel functions between contacting bodies is proposed to naturally serve as the impenetration condition.

Consider a semi-Lagrangian discretization of a continuum by a set of RKPM points, as shown in

Figure 15, with each point carrying nodal volume A_I, mass m_I, kernel function $\varphi_a(\boldsymbol{x}-\boldsymbol{x}_I)$, and state and field variables. The interaction between RKPM points (

Figure 15c) via the overlap of kernel supports induces a stress

$$\bar{\boldsymbol{\tau}}(\boldsymbol{x}) = \sum_I \boldsymbol{D}\bar{\boldsymbol{B}}_I(\boldsymbol{x})\boldsymbol{d}_I \tag{32}$$

where $\boldsymbol{D}$ is the material response tensor mimicking Coulomb's contact frictional law, $\bar{\boldsymbol{B}}_I(\boldsymbol{x}_J)$ is the smoothed gradient of the shape function if

SCNI or SNNI is introduced for the domain integration, and consequently the internal force acting on a point *I* is

$$\boldsymbol{f}_I = \sum_{J \in N_I} \bar{\boldsymbol{B}}_I^T \left(\boldsymbol{x}_J \right) \bar{\boldsymbol{\tau}} \left(\boldsymbol{x}_J \right) A_J \tag{33}$$

where $N_I = \left\{ J \mid \varphi_a \left(\boldsymbol{x}_I - \boldsymbol{x}_J \right) \neq 0, \boldsymbol{x}_J \in G_I \right\}$ is the set containing neighbors of point *I*, and G_I is the group of points located in the body that contains point *I*. In the preceding equation, the total force acting on point *I* is obtained by summing up all pair interactions between point *I* and its neighbors. This property is applied to the interaction between contacting bodies as follows.

$$\boldsymbol{f}_I = \sum_{J \in N_I^C} \bar{\boldsymbol{B}}_I^T \left(\boldsymbol{x}_J \right) \bar{\boldsymbol{\tau}} \left(\boldsymbol{x}_J \right) A_J \tag{34}$$

where $N_I^C = \left\{ J \mid \varphi_a \left(\boldsymbol{x}_I - \boldsymbol{x}_J \right) \neq 0, \boldsymbol{x}_J \in G_I \text{ or } G_I^* \right\}$,
$G_I^* = \left\{ J \mid \boldsymbol{x}_J \notin G_I, \boldsymbol{n}_{IJ} \cdot \bar{\boldsymbol{\tau}} \left(\boldsymbol{x}_J \right) \cdot \boldsymbol{n}_{IJ} < 0 \right\}$, and $\boldsymbol{n}_{IJ} = \left(\boldsymbol{x}_I - \boldsymbol{x}_J \right) / \left\| \boldsymbol{x}_I - \boldsymbol{x}_J \right\|$ is the unit vector pointing from point *J* to *I*. In this approach, when two bodies are approaching each other, the pair-wise interactions due to overlapping kernel functions serve as a natural impenetration condition as shown in

Figure 15d. The radius of kernel support determines the numerical length scale in the normal contact. The stick-and-slip conditions can be calculated based on the tangential stress $\boldsymbol{n} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{t}$ in the contact processing zone. Since the contacting bodies exhibit high-velocity gradients across the contact surface, stability conditions are crucial for the kernel contact algorithm to be stable (Guan et al. 2011).

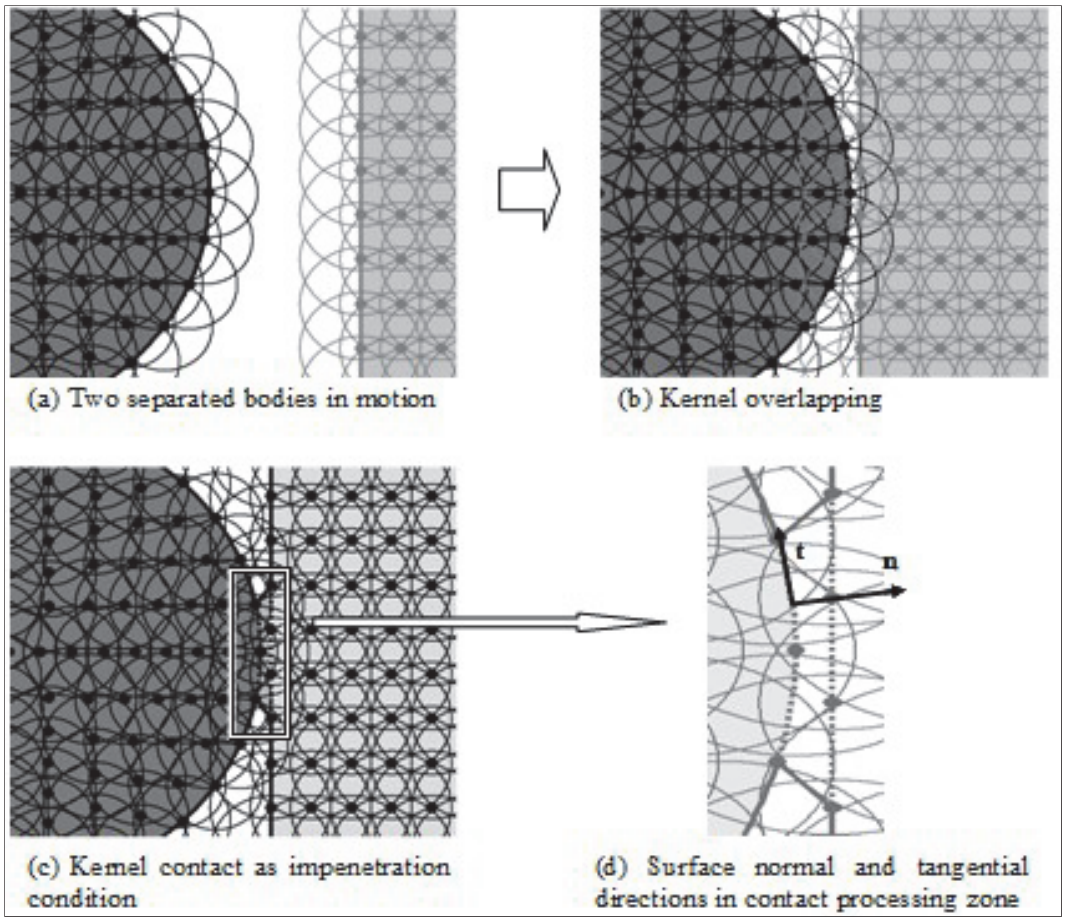


Figure 15. Kernel contact algorithm by kernel interaction between contacting bodies.

Level set algorithm for determination of surface normal

The determination of contact surface is crucial but, on the other hand, difficult to obtain in the kernel algorithm, owing to the point-based discretization of contacting bodies in the RKPM. An oversimplified estimation of surface normal, such as the directional vectors between nodes, leads to an inaccurate contact force calculation. In this section, a level set method to estimate the contact surface normal under the RK-based kernel contact framework is introduced.

Consider a level set function constructed by

$$\phi(\mathbf{x}) = \sum_I \Psi_I(\mathbf{x}) C_I, \quad I \in G_1 \cup G_2 \tag{35}$$

where G_K is the group of points contained in body K and C_I is the level set nodal value associated with the RK shape function Ψ_I (

Figure 16a). The level set nodal value is defined as

$$C_I = \begin{cases} 1 & if\ I \in G_1 \\ -1 & if\ I \in G_2 \end{cases} \tag{36}$$

The level set function in Equation 35 is used to obtain a zero level set between these two bodies, which serves as the contact surface, as can be seen in

Figure 16b-c. The contact surface outward normal, shown in

Figure 16c, then can be obtained by

$$\mathbf{n} = -\nabla\phi(\mathbf{x}) / \left\|\nabla\phi(\mathbf{x})\right\| \tag{37}$$

The gradient operator in Equation 37 can be replaced by the smoothed gradient operator described for SCNI/SNNI integration. The contact force can be therefore applied to the contact points following the frictional kernel contact algorithm described in the previous section.

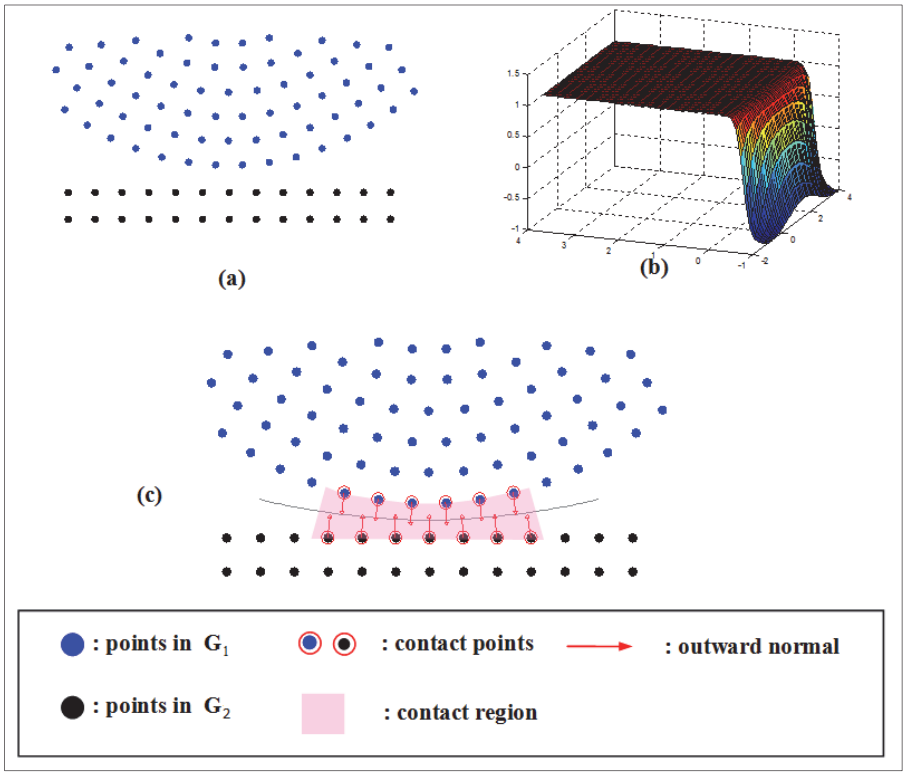


Figure 16. Level set algorithm to identify contact nodes and obtain normal vector of the contact surface.

6 Example Problems

Wave propagation problem

This problem is analyzed using Lagrangian RKPM with SCNI domain integration. The first example problem is the wave propagation in an elastic rod, as illustrated in

Figure 17. The bar has the dimensions of 5 m by 5 m by 0.02 m and is fixed at the right end. An initial velocity $V_0 = 1000$ m/sec in the *x*-direction is prescribed. The material properties are Young's modulus $E = 29$ MPa, Poisson's ratio ν=0, and mass density $\rho = 0.73395 \times 10^{-3}$ kg/m^3. The rod is discretized into 1,004 RKPM particles (251 particles in the *x*-direction and four particles on each *y*-*z* cross-section). LSEMI is set to be 0 for RKPM Lagrangian modeling using *NMAP*.

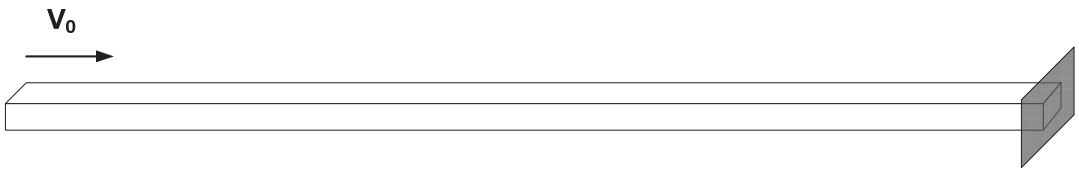


Figure 17. Geometry of elastic wave propagation problem.

Figures 18 and 19 show the time-histories of displacement and velocity at the free end and at the midpoint of the elastic bar, respectively. Two node groups are defined in *NMAP_Input.dat,* as demonstrated in Figure 20. Node Group 1 (Node 1 to Node 1,000) is used to prescribe an initial velocity, and Node Group 2 (Node 1 to Node 1,004) is used to define the material set and body set for the elastic bar. Portions of all the required input files are shown in Figures 21 through 29. The files *boun.dat* and *data.id*, which define the Voronoi cell information, are provided for the SCNI calculation.

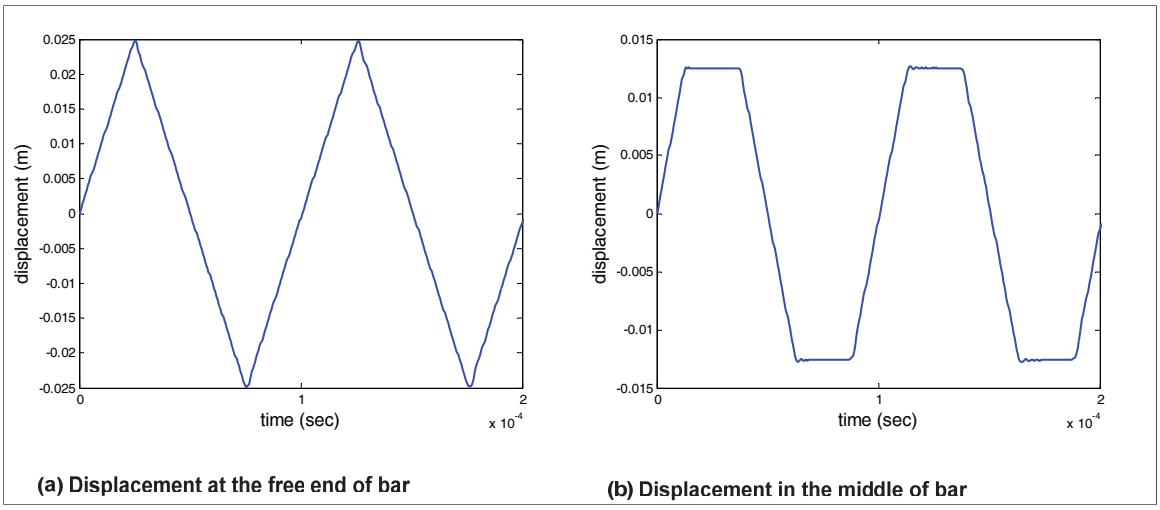


Figure 18. Numerical results of displacement responses for elastic wave propagation problem.

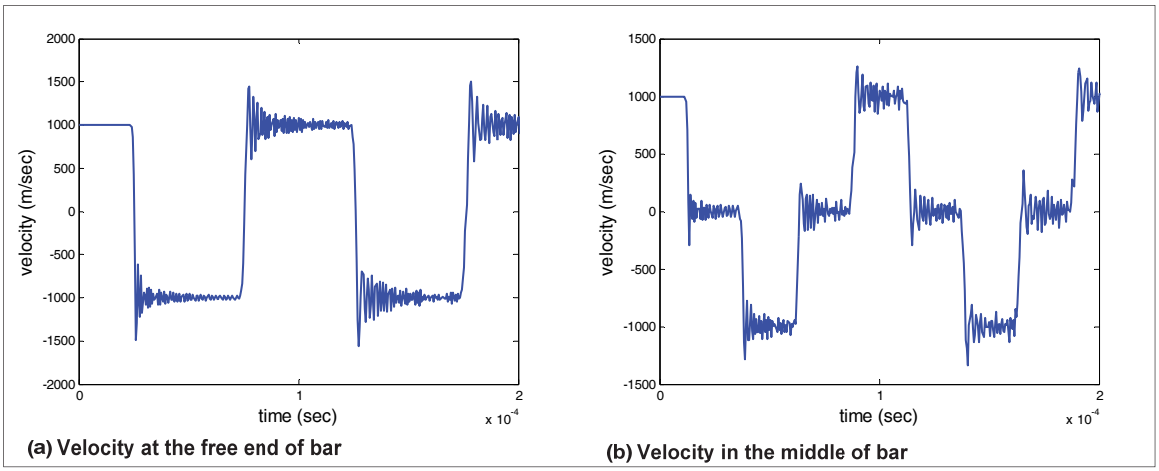


Figure 19. Numerical results of velocity responses for elastic wave propagation problem.

```
* RKPM Parameter
# Basis Func Order[NORDER]       KERNEL TYPE [ISPLINE] SUPPORT SIZE(0:use vales below, 1: auto by code)[IDILA]    Mesh
TYPE(H, T)[CMESH]
       1                    4 1                         H
# [(DCJP(1:3)] used if IDILA = 0
       0.500000E-01 0.500000E-01 0.500000E-01

* Logical Parameters
# [LSEMI] [LFEMRK] GRAVITY[LGRAV] AUTO SUPPORT[LSUPADJ] RESTART[LNEW] (1:new job, 0:restart) LBINARY ICONTACT
 0 0 0 0 1 0
# FREQUENCY OF UPDATING THE SEARCHBOX [IBOXSW] (used when LSEMI is 1)
 200

* Output Control
# [IOutFreq]  [IDISP]       [ISIGEP]       [IECHO] [ISUPP] [IMASS] [IFORCE] [IINTEROUT]
 250           1 0 1 1 1

* Node Group Definition
# Num of Node Groups [NUM_GROUP]
       2
# Group ID    Type (1: Order, 2: Specify) Node_Beg OR # OF NODES    Node_End      Node_Inc [NODE_GROUP_ID]
       1       1             1      1000 1

       2 1 1 1004 1

* Body Set
# Num of Bodies [NUM_BODY]
       1
# Set ID       Node Group SCALING FACTOR FOR DCJP [...] [FRIC_COEFF]
       1              2 1.01

* Material Set
# Num of Sets [NUM_MAT]
       1
# Set ID      Set Type       Node Group [...]
       1       1 2
# Set Info: Parameters
1 29.0000E+6 0.0000E+00 0.73395E-03 0.00000E+00
```

Figure 20. *NMAP_Input.dat* input file for elastic wave propagation problem.

```
* Time Integration
# Time_Begin Time_End Time_Incr [TIMS(1:3)]
 0.000000E+00 5.000E-04 0.2000E-08
# [Gamma] >= 0.5
       0.500000E+00
# fCurve(1)          fCurve(2) [FCURVE(1:2)] change to 0 0 later
       0.000000E+00 0
# DCurve(1)          DCurve(2) [DCURVE(1:2)] change to 0 0 later
       0.000000E+00 0
# GRAVITY DIRECTION [G_DIR(1:3)]
 0 0 0

* Initial Displacement
# Num of Initial Displacement Set [Num_IDIS]
       0
# Set ID      Node Group    DISPI(1:3)


* Initial Velocity
# Num of Initial Velocity Set [Num_Vel]
       1
# Set ID      Node Group    VELI(1:3)
 1             1            1000 0 0

* End
```

Figure 20 (continued). *NMAP_Input.dat* input file for elastic wave propagation problem.

```
1004
1 0.00000E+00 0.00000E+00 0.00000E+00
2 0.50000E-02 0.00000E+00 0.00000E+00
3 0.00000E+00 0.50000E-02 0.00000E+00
4 0.50000E-02 0.50000E-02 0.00000E+00
5 0.00000E+00 0.00000E+00 0.50000E-02
6 0.50000E-02 0.00000E+00 0.50000E-02
7 0.00000E+00 0.50000E-02 0.50000E-02
8 0.50000E-02 0.50000E-02 0.50000E-02
9 0.10000E-01 0.00000E+00 0.00000E+00
10 0.10000E-01 0.50000E-02 0.00000E+00
11 0.10000E-01 0.00000E+00 0.50000E-02
12 0.10000E-01 0.50000E-02 0.50000E-02
⋮ ⋮ ⋮ ⋮
1001 0.50000E+01 0.00000E+00 0.00000E+00
1002 0.50000E+01 0.50000E-02 0.00000E+00
1003 0.50000E+01 0.00000E+00 0.50000E-02
1004 0.50000E+01 0.50000E-02 0.50000E-02
18009 4
```

Figure 21. Portion of *Input_coor.dat* input file for elastic wave propagation problem.

```
1 0.10020000E-01 0.10020000E-01 0.10020000E-01
2 0.10020000E-01 0.10020000E-01 0.10020000E-01
3 0.10020000E-01 0.10020000E-01 0.10020000E-01
4 0.10020000E-01 0.10020000E-01 0.10020000E-01
5 0.10020000E-01 0.10020000E-01 0.10020000E-01
⋮ ⋮ ⋮ ⋮
1000 0.10019990E-01 0.10020000E-01 0.10020000E-01
1001 0.10016407E-01 0.10020000E-01 0.10020000E-01
1002 0.10016407E-01 0.10020000E-01 0.10020000E-01
1003 0.10016407E-01 0.10020000E-01 0.10020000E-01
1004 0.10016407E-01 0.10020000E-01 0.10020000E-01
```

Figure 22. Portion of *Input_dila.dat* input file for elastic wave propagation problem.

```
0 1000

1 0.10000000E+04 0.00000000E+00 0.00000000E+00
2 0.10000000E+04 0.00000000E+00 0.00000000E+00
3 0.10000000E+04 0.00000000E+00 0.00000000E+00
4 0.10000000E+04 0.00000000E+00 0.00000000E+00
5 0.10000000E+04 0.00000000E+00 0.00000000E+00
⋮ ⋮ ⋮ ⋮
998 0.10000000E+04 0.00000000E+00 0.00000000E+00
999 0.10000000E+04 0.00000000E+00 0.00000000E+00
1000 0.10000000E+04 0.00000000E+00 0.00000000E+00
```

Figure 23. Portion of *Input_initial.dat* input file for elastic wave propagation problem.

```
1 1 1
2 1 1
3 1 1
4 1 1
5 1 1
⋮ ⋮ ⋮
1000 1 1
1001 1 1
1002 1 1
1003 1 1
1004 1 1
```

Figure 24. Portion of *Input_id.dat* input file for elastic wave propagation problem.

```
0
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
5 5 5 5
⋮ ⋮ ⋮ ⋮
999 999 999 999
1000 1000 1000 1000
1001 0 0 0
1002 0 0 0
1003 0 0 0
1004 0 0 0
```

Figure 25. Portion of *Input_bound.dat* input file for elastic wave propagation problem.

```
0
```

Figure 26. *Input_nforce.dat* input file for elastic wave propagation problem.

```
0.00000E+00 0.00000E+00 0.00000E+00
0.20000E-01 0.00000E+00 0.00000E+00
0.40000E-01 0.00000E+00 0.00000E+00
0.60000E-01 0.00000E+00 0.00000E+00
0.80000E-01 0.00000E+00 0.00000E+00
0.10000E+00 0.00000E+00 0.00000E+00
0.12000E+00 0.00000E+00 0.00000E+00
0.14000E+00 0.00000E+00 0.00000E+00
0.16000E+00 0.00000E+00 0.00000E+00
0.18000E+00 0.00000E+00 0.00000E+00
⋮ ⋮ ⋮
0.49900E+01 0.20000E-01 0.00000E+00
0.49900E+01 0.10000E-01 0.00000E+00
0.49900E+01 0.00000E+00 0.10000E-01
0.50000E+01 0.10000E-01 0.10000E-01
0.49900E+01 0.20000E-01 0.10000E-01
0.49900E+01 0.00000E+00 0.20000E-01
0.50000E+01 0.10000E-01 0.20000E-01
```

```
0.49900E+01 0.20000E-01 0.20000E-01
0.49900E+01 0.10000E-01 0.20000E-01 (LINE: 4509)
```

Figure 27. Portion of *boun.dat* input file for elastic wave propagation problem.

```
0
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
5 5 5 5
6 6 6 6
7 7 7 7
8 8 8 8
⋮ ⋮ ⋮ ⋮
997 997 997 997
998 998 998 998
999 999 999 999
1000 1000 1000 1000
1001 0 0 0
1002 0 0 0
1003 0 0 0
1004 0 0 0
```

Figure 28. Portion of *Input_bound.dat* input file for elastic wave propagation problem.

```
1 0.100000E-05 6
1 0.10000E+01 0.00000E+00 0.00000E+00 0.10000E-03 1010 1014 1009 1015
1 -0.10000E+01 0.00000E+00 0.00000E+00 0.10000E-03 1 1005 1018 1013
1 0.00000E+00 0.10000E+01 0.00000E+00 0.10000E-03 1014 1013 1018 1009
1 0.00000E+00 -0.10000E+01 0.00000E+00 0.10000E-03 1 1010 1015 1005
1 0.00000E+00 0.00000E+00 0.10000E+01 0.10000E-03 1005 1015 1009 1018
1 0.00000E+00 0.00000E+00 -0.10000E+01 0.10000E-03 1 1013 1014 1010
2 0.200000E-05 10
2 -0.10000E+01 0.00000E+00 0.00000E+00 0.10000E-03 1010 1015 1009 1014
2 0.00000E+00 0.10000E+01 0.00000E+00 0.10000E-03 1011 1014 1009 1016
2 0.00000E+00 -0.10000E+01 0.00000E+00 0.10000E-03 1010 2 1006 1015
2 0.00000E+00 0.00000E+00 0.10000E+01 0.10000E-03 1015 1006 1016 1009
2 0.00000E+00 0.00000E+00 -0.10000E+01 0.10000E-03 1010 1014 1011 2
2 0.10000E+01 0.00000E+00 0.00000E+00 0.10000E-03 1027 1030 1026 1031
2 0.00000E+00 0.10000E+01 0.00000E+00 0.10000E-03 1030 1011 1016 1026
2 0.00000E+00 -0.10000E+01 0.00000E+00 0.10000E-03 2 1027 1031 1006
2 0.00000E+00 0.00000E+00 0.10000E+01 0.10000E-03 1006 1031 1026 1016
2 0.00000E+00 0.00000E+00 -0.10000E+01 0.10000E-03 2 1011 1030 1027
⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮
1002 0.100000E-05 6
1002 0.10000E+01 0.00000E+00 0.00000E+00 0.10000E-03 4500 1002 4497 4504
1002 -0.10000E+01 0.00000E+00 0.00000E+00 0.10000E-03 4502 4498 4505 4501
1002 0.00000E+00 0.10000E+01 0.00000E+00 0.10000E-03 1002 4501 4505 4497
1002 0.00000E+00 -0.10000E+01 0.00000E+00 0.10000E-03 4502 4500 4504 4498
1002 0.00000E+00 0.00000E+00 0.10000E+01 0.10000E-03 4498 4504 4497 4505
1002 0.00000E+00 0.00000E+00 -0.10000E+01 0.10000E-03 4502 4501 1002 4500
1003 0.100000E-05 6
1003 0.10000E+01 0.00000E+00 0.00000E+00 0.10000E-03 4496 4504 4507 1003
1003 -0.10000E+01 0.00000E+00 0.00000E+00 0.10000E-03 4503 4506 4509 4498
1003 0.00000E+00 0.10000E+01 0.00000E+00 0.10000E-03 4504 4498 4509 4507
1003 0.00000E+00 -0.10000E+01 0.00000E+00 0.10000E-03 4503 4496 1003 4506
1003 0.00000E+00 0.00000E+00 0.10000E+01 0.10000E-03 4506 1003 4507 4509
1003 0.00000E+00 0.00000E+00 -0.10000E+01 0.10000E-03 4503 4498 4504 4496
1004 0.100000E-05 6
1004 0.10000E+01 0.00000E+00 0.00000E+00 0.10000E-03 4504 4497 1004 4507
1004 -0.10000E+01 0.00000E+00 0.00000E+00 0.10000E-03 4498 4509 4508 4505
1004 0.00000E+00 0.10000E+01 0.00000E+00 0.10000E-03 4497 4505 4508 1004
1004 0.00000E+00 -0.10000E+01 0.00000E+00 0.10000E-03 4498 4504 4507 4509
1004 0.00000E+00 0.00000E+00 0.10000E+01 0.10000E-03 4509 4507 1004 4508
1004 0.00000E+00 0.00000E+00 -0.10000E+01 0.10000E-03 4498 4505 4497 4504
total number of edges = 10008
total vol. = 2.000000000000034E-003
```

Figure 29. Portion of *data.id* input file for elastic wave propagation problem.

Cylindrical aluminum bar impacting on a rigid surface

This problem is modeled by *NMAP* using semi-Lagrangian RKPM formulation with SNNI integration method and frictional kernel contact algorithm. This classical impact problem with available experimental and numerical results (Taylor 1948; Wilkins and Guinan 1973) is used to test the performance of the proposed contact algorithm. The initial radius and initial height of the aluminum cylindrical bar are 0.391 cm and 2.346 cm, respectively. The material properties of the cylinder are density $\rho = 2700$ kg/m^3, Young's modulus $E = 78.2$ GPa, Poisson's ratio $\nu = 0.3$, and J2 plasticity with yield strength $\sigma_Y = 0.29$ GPa. The initial impact velocity is 373 m/sec, and the rigid surface is assumed to be frictionless.

Perfect plasticity and strain-hardening elasto-plasticity with the following hardening rules are considered for this problem.

$$H(\bar{e}^p) = 0 \tag{38}$$

$$K(\bar{e}^p) = \sigma_T(1 + 125\bar{e}^p)^{0.1} \tag{39}$$

where $\bar{e}^p$ is the effective plastic strain, and H and K are the plastic modulus and yield stress, respectively (Chen et al. 1996).

Both the aluminum bar and the rigid plate are discretized by the RKPM particles in three-dimensions (29,788 RKPM particles), as illustrated in

Figure 30b. The contact between the aluminum bar and the rigid wall is handled by the frictional kernel contact algorithm, and the problem is modeled under the semi-Lagrangian SNNI framework.

Table 5 summarizes the comparison of deformed heights and radii between RKPM predictions and experimental measurements. The deformed configurations of the aluminum bar at different time frames obtained by the semi-Lagrangian RKPM with SNNI calculation are shown in

Figure 31. The required input files associated with this example are given in Figures 32 through 38.

Table 5. Comparison of deformed geometries for Taylor bar impact problem.

	RKPM Lagrangian[a] (Chen et al. 1996)	RKPM Semi-Lagrangian SNNI	Experiment (Wilkins and Guinan 1973)
Height (cm)	1.645	1.642	1.651
Radius (cm)	0.837	0.819	NA
[a] Axisymmetric model with 11×31 particles was analyzed in Chen et al. (1996).			

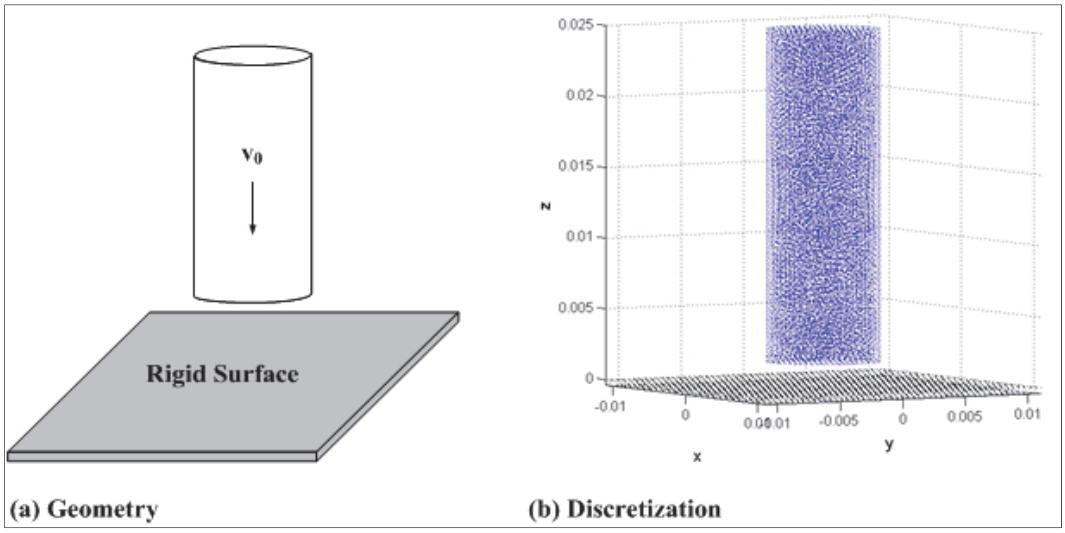


Figure 30. Schematic of Taylor bar impact problem and the corresponding RKPM discretization.

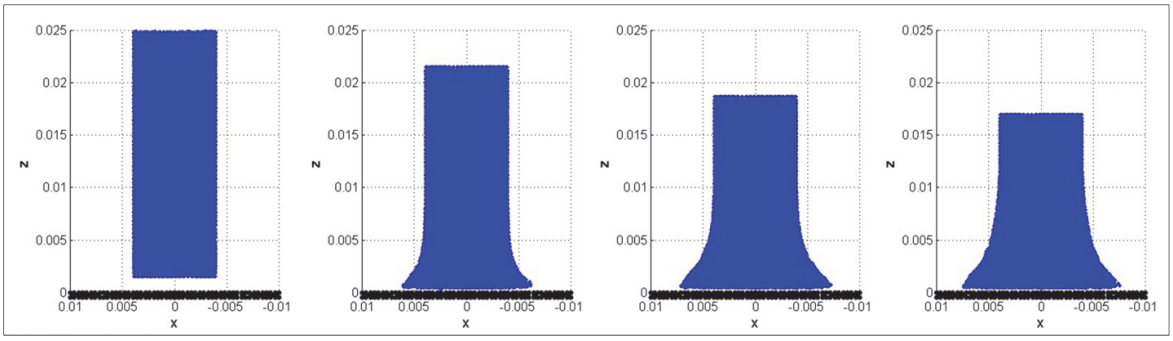


Figure 31. Cylindrical impact deformations predicted by the semi-Lagrangian SNNI RKPM.

```
* RKPM Parameter
# Basis Func Order[NORDER]       KERNEL TYPE [ISPLINE] SUPPORT SIZE(0:use vales below, 1: auto by code)[IDILA]    Mesh
TYPE(H, T)[CMESH]
       1                    4 1                              T
# [(DCJP(1:3)] used if IDILA = 0
       0.123000E-01 0.123000E-01 0.123000E-01

* Logical Parameters
# [LSEMI] [LFEMRK] GRAVITY[LGRAV] AUTO SUPPORT[LSUPADJ] RESTART[LNEW] (1:new job, 0:restart) LBINARY ICONTACT
 1 0 0 0 1 0
# FREQUENCY OF UPDATING THE SEARCHBOX [IBOXSW] (used when LSEMI is 1)
 50

* Output Control
# [IOutFreq]  [IDISP]       [ISIGEP]       [IECHO] [ISUPP] [IMASS] [IFORCE] [IINTEROUT]
 100           1 1 1 1 1

* Node Group Definition
# Num of Node Groups [NUM_GROUP]
       2
# Group ID    Type (1: Order, 2: Specify) Node_Beg OR # OF NODES    Node_End      Node_Inc [NODE_GROUP_ID]
       1       1                           1     24493   1

 2 1 24494 29788 1

* Body Set
# Num of Bodies [NUM_BODY]
       2
# Set ID       Node Group SCALING FACTOR FOR DCJP [...] [FRIC_COEFF]
       1             1 1.6501
       2             2 1.6501

* Material Set
# Num of Sets [NUM_MAT]
       2
# Set ID      Set Type       Node Group [...]
       1       3 1
       2       1 2
# Set Info: Parameters
1 78.2000E+9 0.30000E+00 -0.124000E+03 0.100000E+01 0.290000E+09 0.100000E+00 0.001000E+00 0.270000E+04 0.000000E-03
2 78.2000E+9 0.30000E+00 0.270000E+04 0.000000E-03
```

Figure 32. *NMAP_Input.dat* input file for Taylor bar problem.

```
* Time Integration
# Time_Begin Time_End Time_Incr [TIMS(1:3)]
 0.000000E+00 4.5E-05 0.2E-8
# [Gamma] >= 0.5
       0.500000E+00
# fCurve(1)         fCurve(2) [FCURVE(1:2)] change to 0 0 later
       0.000000E+00 0
# DCurve(1)         DCurve(2) [DCURVE(1:2)] change to 0 0 later
       0.000000E+00 0.0
# GRAVITY DIRECTION [G_DIR(1:3)]
 0 0 1

* Initial Displacement
# Num of Initial Displacement Set [Num_IDIS]
       0
# Set ID     Node Group    DISPI(1:3)


* Initial Velocity
# Num of Initial Velocity Set [Num_Vel]
       1
# Set ID     Node Group    VELI(1:3) change to 30000 0 0 later
 1 1 0 0 -373.0

* End
```

Figure 32 (continued). *NMAP_Input.dat* input file for Taylor bar problem.

```
29788
1 0.98226219E-03 0.17492393E-02 0.32814830E-02
2 -0.77495060E-03 0.12723466E-02 0.10346784E-01
3 0.24641986E-03 0.18783344E-02 0.78992927E-02
4 0.96762925E-03 0.11523374E-02 0.13219517E-01
5 0.98833733E-03 0.17074732E-02 0.72545763E-02
⋮ ⋮ ⋮ ⋮
29784 0.88012069E-02 0.95222807E-02 -0.16000001E-02
29785 0.92010358E-02 0.92890700E-02 -0.16000001E-02
29786 0.96006021E-02 0.90509858E-02 -0.16000001E-02
29787 0.92011997E-02 0.97141890E-02 -0.16000001E-02
29788 0.96007409E-02 0.94779842E-02 -0.16000001E-02
0 0
```

Figure 33. Portion of *Input_coor.dat* input file for Taylor bar problem.

```
1 0.22090696E-03 0.19174880E-03 0.24357120E-03
2 0.12922882E-03 0.18221856E-03 0.26846720E-03
3 0.21271541E-03 0.28743266E-03 0.18564955E-03
4 0.20697191E-03 0.26653604E-03 0.26834097E-03
5 0.27485264E-03 0.18793673E-03 0.19403576E-03
⋮ ⋮ ⋮ ⋮
29785 0.20073093E-03 0.21340971E-03 0.20080001E-03
29786 0.20060470E-03 0.21435318E-03 0.20080001E-03
29787 0.20092028E-03 0.21340971E-03 0.20080001E-03
29788 0.20065192E-03 0.26205184E-03 0.20080001E-03
```

Figure 34. Portion of *Input_dila.dat* input file for Taylor bar problem.

```
0 24493

1 0.00000000E+00 0.00000000E+00 -0.37300000E+03
2 0.00000000E+00 0.00000000E+00 -0.37300000E+03
3 0.00000000E+00 0.00000000E+00 -0.37300000E+03
4 0.00000000E+00 0.00000000E+00 -0.37300000E+03
5 0.00000000E+00 0.00000000E+00 -0.37300000E+03
6 0.00000000E+00 0.00000000E+00 -0.37300000E+03
⋮ ⋮ ⋮ ⋮
24489 0.00000000E+00 0.00000000E+00 -0.37300000E+03
24490 0.00000000E+00 0.00000000E+00 -0.37300000E+03
24491 0.00000000E+00 0.00000000E+00 -0.37300000E+03
24492 0.00000000E+00 0.00000000E+00 -0.37300000E+03
24493 0.00000000E+00 0.00000000E+00 -0.37300000E+03
```

Figure 35. Portion of *Input_initial.dat* input file for Taylor bar problem.

```
1 1 1
2 1 1
3 1 1
4 1 1
5 1 1
6 1 1
⋮ ⋮ ⋮
    29785 2 2
    29786 2 2
    29787 2 2
    29788 2 2
```

Figure 36. Portion of *Input_id.dat* input file for Taylor bar problem.

```
0
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
5 5 5 5
⋮ ⋮ ⋮ ⋮
    29784 0 0 0
    29785 0 0 0
    29786 0 0 0
    29787 0 0 0
    29788 0 0 0
```

Figure 37. Portion of *Input_bound.dat* input file for Taylor bar problem.

```
0
```

Figure 38. *Input_nforce.dat* input file for Taylor bar problem.

CorTuf thin panel perforation problem (SNNI and semi-Lagrangian approximation)

A thin panel perforation (V&V Run 2 in Chen et al. 2011) problem, as depicted in

Figure 39,was modeled using *NMAP*. The 129-grain spherical projectile with 12.7-mm diam was made of hardened S2 tool steel and modeled by J2 plasticity model with material parameters listed in Table 6. The target concrete panel with dimensions of 0.0127 m by 0.3048 m by 0.3048 m was made of fiber-reinforced CorTuf concrete and modeled by the multiscale AFC model. The corresponding material constants are given in Table 7. The projectile had an initial velocity of 114.91 m/sec, and the panel was initially at rest and fixed on all edges. LSEMI was set to be 1 for semi-Lagrangian RKPM-SNNI calculation.

The problem was discretized into 150,355 RKPM particles. Two node groups were defined in *NMAP_Input.dat*. Node Group 1 (Node 1 to Node 1517) was for the steel projectile, and Node Group 2 (Node 1518 to Node 150,355) was for the target panel. Portions of the required input files

are shown in Figures 42 through 48. The files *boun.dat* and *data.id* are not needed for semi-Lagrangian modeling.

Figure 40 shows the numerical prediction of the projectile's velocity history compared with the experimentally measured exit velocity (marked as star).

Figure 41a and 41b demonstrate the shear damage and tensile damage patterns on the impact and exit faces, respectively.

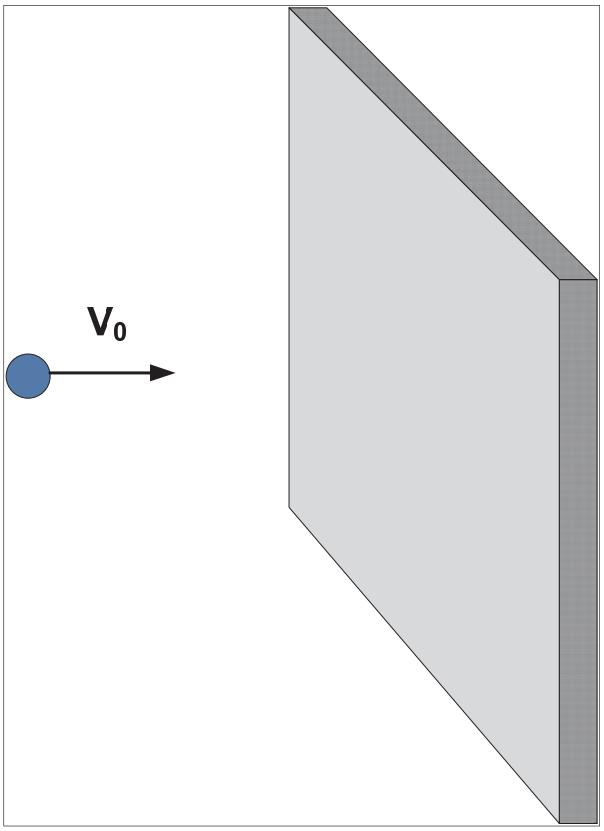


Figure 39. Geometry of CorTuf thin panel perforation problem.

Table 6. Spherical projectile properties.

Property	Value
Young's modulus, E	200 GPa
Poisson's ratio, *v*	0.26

Yield stress, σ_y[a]	2400 MPa
Hardening modulus, H	2500 MPa
DAMP (mass proportional damping)	0.0001
Mass density	7806 kg/m^3
[a] Dynamic Increase Factor (DIF) = 1.2.	

Table 7. AFC model parameters for CorTuf panel.

G (shear modulus)	18.457 GPa
C1	1,016.3 MPa
C2	908.65 MPa
C3	0.0125
C4	0.10382
C5	792.89 MPa
Q1	artificial bulk viscosity not used
Q2	artificial bulk viscosity not used
PMIN	6.8947 MPa
C6	172.37 MPa
C7	0.00781
C	7,919.2 MPa
D	-29.205 GPa
S	187.10 GPa
C9	77.958 GPa
C10	0.24863
D1	4.0597×10-10 Pa-1
AN	1.7345×10-9 Pa-1
TXETXCR	0.625
PRECRIT	0.177×1022
HMIN	1
RHO (density) [a]	2,267.4 kg/m^3
DAMP (mass proportional damping)	0.0001
FC (tensile strength for MIDM)	6.8947 MPa
[a] 2,267.4 kg/m^3 was the value used in the model; density reported by ERDC was 2,557 kg/m^3.	

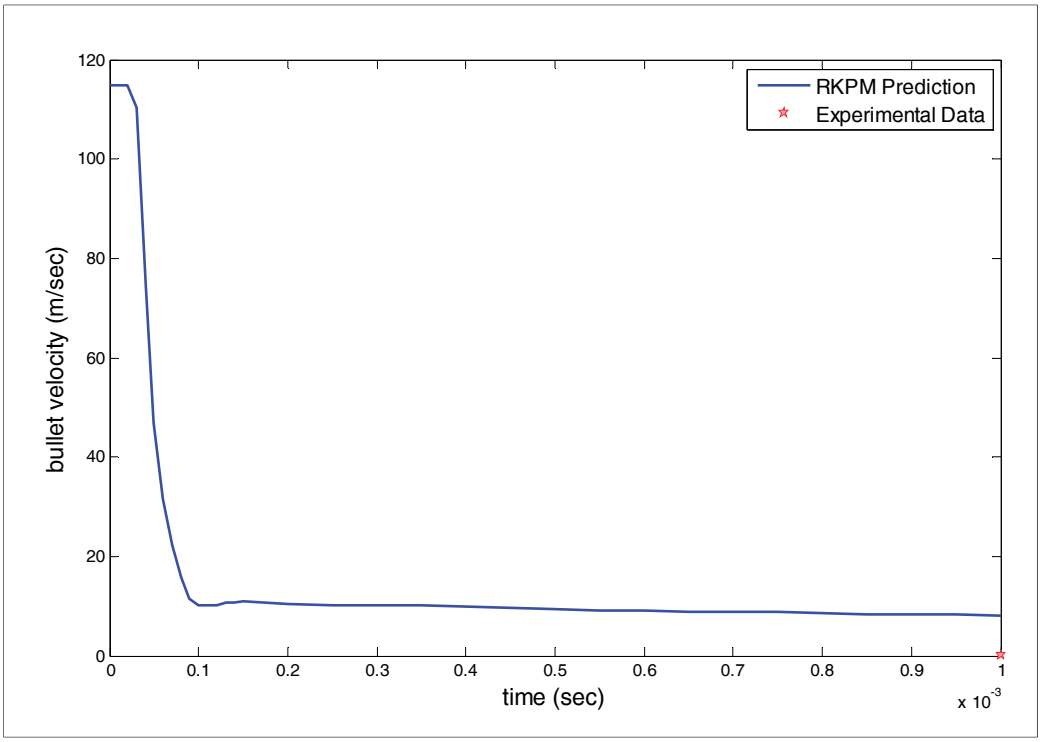
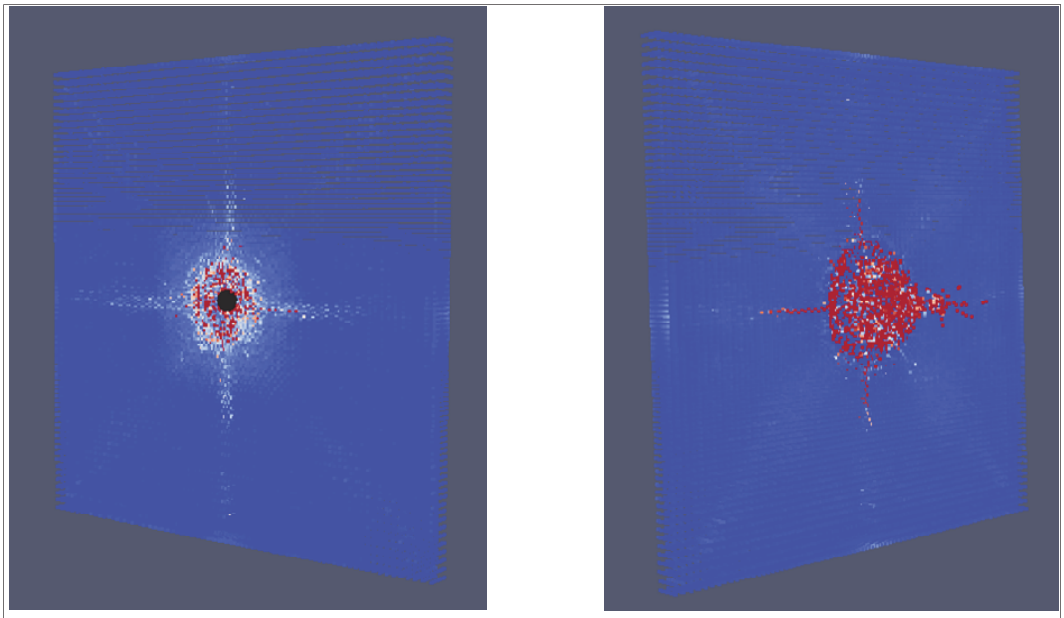


Figure 40. Bullet velocity history of CorTuf thin panel perforation problem.



(a) Shear damage (impact face) (b) Tensile damage (exit face)

Figure 41. Damage patterns of CorTuf thin panel perforation problem.

```
* RKPM Parameter
# Basis Func Order[NORDER]  KERNEL TYPE [ISPLINE] SUPPORT SIZE(0:use vales below, 1: auto by code)[IDILA]  Mesh TYPE(H,
T)[CMESH]
       0                     4 1                                  H
# [(DCJP(1:3)] used if IDILA = 0
       0.500000E-01 0.500000E-01 0.500000E-01

* Logical Parameters
# [LSEMI] [LFEMRK] GRAVITY[LGRAV] AUTO SUPPORT[LSUPADJ] RESTART[LNEW] (1:new job, 0:restart) LBINARY ICONTACT
 1 0 0 0 1 0
# FREQUENCY OF UPDATING THE SEARCHBOX [IBOXSW] (used when LSEMI is 1)
 50

* Output Control
# [IOutFreq]   [IDISP][ISIGEP]       [IECHO] [ISUPP] [IMASS] [IFORCE] [IINTEROUT]
 500           1 0 1 1 1

* Node Group Definition
# Num of Node Groups [NUM_GROUP]
       2
# Group ID    Type (1: Order, 2: Specify) Node_Beg OR # OF NODES      Node_End      Node_Inc [NODE_GROUP_ID]
       1       1                                  1       1517   1

       2 1           1518 150355 1

* Body Set
# Num of Bodies [NUM_BODY]
       2
# Set ID       Node Group SCALING FACTOR FOR DCJP [...] [FRIC_COEFF]
       1             1 1.05
       2             2 1.01

* Material Set
# Num of Sets [NUM_MAT]
       2
# Set ID      Set Type       Node Group [...]
       1       3 1
       2 5 2
# Set Info: Parameters
1 0.20000E+12 0.26000E+00 0.00000E+00 0.25000E+10 0.24000E+10 1.00000E+00 0.10000E-05 0.780573E+04 0.00010E+00

2 0.26770E+07 0.14741E+06 0.13179E+06 0.12500E-01 0.10382E+00 0.11500E+06 0.20000E+00 0.40000E+01 0.10000E+04
 0.25000E+05 0.78100E-02 0.11486E+07 -0.42360E+07 0.27137E+08 0.11307E+08 0.24863E+00 0.27991E-05 0.11959E-04
 0.62500E+00 0.17700E+22 0.10000E+01 0.22674E+04 0.00010E+00 0.12000E+04 1 1 0.68927E+04
```

Figure 42. *NMAP_Input.dat* input file for CorTuf thin panel perforation problem.

```
* Time Integration
# Time_Begin Time_End Time_Incr [TIMS(1:3)]
 0.000000E+00 0.100100E-02 2.0000E-08
# [Gamma] >= 0.5
       0.500000E+00
# fCurve(1)         fCurve(2) [FCURVE(1:2)] change to 0 0 later
       0.000000E+00 0
# DCurve(1)         DCurve(2) [DCURVE(1:2)] change to 0 0 later
       0.000000E+00 0
# GRAVITY DIRECTION [G_DIR(1:3)]
 0 0 1

* Initial Displacement
# Num of Initial Displacement Set [Num_IDIS]
       0
# Set ID     Node Group    DISPI(1:3)


* Initial Velocity
# Num of Initial Velocity Set [Num_Vel]
       1
# Set ID     Node Group    VELI(1:3) change to 30000 0 0 later
 1            1           114.91 0 0

* End
```

Figure 42 (continued). *NMAP_Input.dat* input file for CorTuf thin panel perforation problem.

```
150354
1 -0.41485E-02 0.00000E+00 -0.25578E-09
2 -0.42205E-02 0.91538E-03 -0.25578E-09
3 -0.44349E-02 0.18082E-02 -0.25578E-09
4 -0.47862E-02 0.26565E-02 -0.25578E-09
5 -0.52660E-02 0.34394E-02 -0.25578E-09
⋮ ⋮ ⋮ ⋮
150350 0.14552E-01 -0.12973E+00 -0.14975E+00
150351 0.14552E-01 -0.13454E+00 -0.14975E+00
150352 0.14552E-01 -0.13948E+00 -0.14975E+00
150353 0.14552E-01 -0.14455E+00 -0.14975E+00
150354 0.14552E-01 -0.14975E+00 -0.14975E+00
0 0
```

Figure 43. Portion of *Input_coor.dat* input file for CorTuf thin panel perforation problem.

```
1 5.80000E-004 5.80000E-004 5.80000E-004
2 5.80000E-004 5.80000E-004 5.80000E-004
3 5.80000E-004 5.80000E-004 5.80000E-004
4 5.80000E-004 5.80000E-004 5.80000E-004
5 5.80000E-004 5.80000E-004 5.80000E-004
⋮ ⋮ ⋮ ⋮
150350 0.66146E-03 0.24459E-02 0.26496E-02
150351 0.66146E-03 0.25120E-02 0.26496E-02
150352 0.66146E-03 0.25799E-02 0.26496E-02
150353 0.66146E-03 0.26496E-02 0.26496E-02
150354 0.66146E-03 0.26496E-02 0.26496E-02
```

Figure 44. Portion of *Input_dila.dat* input file for CorTuf thin panel perforation problem.

```
0 1517

1 0.11491E+03 0.00000E+00 0.00000E+00
2 0.11491E+03 0.00000E+00 0.00000E+00
3 0.11491E+03 0.00000E+00 0.00000E+00
4 0.11491E+03 0.00000E+00 0.00000E+00
5 0.11491E+03 0.00000E+00 0.00000E+00
6 0.11491E+03 0.00000E+00 0.00000E+00
⋮ ⋮ ⋮ ⋮
1514 0.11491E+03 0.00000E+00 0.00000E+00
1515 0.11491E+03 0.00000E+00 0.00000E+00
1516 0.11491E+03 0.00000E+00 0.00000E+00
1517 0.11491E+03 0.00000E+00 0.00000E+00
```

Figure 45. Portion of *Input_initial.dat* input file *for* CorTuf thin panel perforation problem.

```
1 1 1
2 1 1
3 1 1
4 1 1
5 1 1
6 1 1
⋮ ⋮ ⋮
150351 2 2
150352 2 2
150353 2 2
150354 2 2
```

Figure 46. Portion of *Input_id.dat* input file for CorTuf thin panel perforation problem.

```
0
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
5 5 5 5
⋮ ⋮ ⋮ ⋮
150349 0 0 0
150350 0 0 0
150351 0 0 0
150352 0 0 0
150353 0 0 0
150354 0 0 0
```

Figure 47. Portion of *Input_bound.dat* input file for CorTuf thin panel perforation problem.

```
0
```

Figure 48. *Input_nforce.dat* input file for CorTuf thin panel perforation problem.

References

Adley, M. D., A. O. Frank, K. T. Danielson, S. A. Akers, and J. L. O'Daniel. 2010. *The Advanced Fundamental Concrete (AFC) model.* Technical Report ERDC/GSL TR-10-51. Vicksburg, MS: U.S. Army Engineer Research and Development Center.

Belytschko, T., Y. Y. Lu, and L. Gu. 1994. Element-free Galerkin methods. *International Journal of Numerical Methods in Engineering,* 37:229-256.

Chen, J. S., C. H. Pan, C. T. Tu, and W. K. Liu. 1996. Reproducing kernel particle methods for large deformation analysis of non-linear structures. *Computer Methods in Applied Mechanics and Engineering* 139:195-227.

Chen, J. S., and H. P. Wang. 2000. Some recent improvements in meshfree methods for incompressible finite elasticity boundary value problems with contact. *Computational Mechanics* 25:137-156.

Chen, J. S., C. T. Wu, S. Yoon, and Y. You. 2001. A stabilized conforming nodal integration for Galerkin meshfree methods. *International Journal of Numerical Methods in Engineering* 50:435-466.

Chen, J. S., S. Yoon, and C. T. Wu. 2002. Nonlinear version of stabilized conforming nodal integration for Galerkin meshfree methods. *International Journal of Numerical Methods in Engineering* 53:2587-2615.

Chen, J. S., W. Hu, M. Puso, Y. Wu, and X. Zhang. 2006. Strain smoothing for stabilization and regularization of Galerkin meshfree method. *Lecture Notes in Computational Science and Engineering* 57:57-76.

Chen, J. S., and Y. Wu. 2007. Stability in Lagrangian and semi-Lagrangian reproducing kernel discretizations using nodal integration in nonlinear solid mechanics. In *Computational methods in applied sciences*, ed. V. M. A. Leitao, C. J. S. Alves, and C. A. Duarte, 55-77. Springer.

Chen, J. S., P. C. Guan, L. Du, Y. C. Lin, and C. H. Lee. 2009. *A multiscale meshfree approach for modeling fragment penetration into ultra-high strength concrete: Part I.* Project Technical Report. Los Angeles, CA: University of California, Los Angeles.

Chen, J. S., S. W. Chi, C. H. Lee, S. P. Lin, C. Marodon, M. J. Roth and T. R. Slawson. 2011. *A multiscale meshfree approach for modeling fragment penetration into ultra-high strength concrete.* Technical Report ERDC/GSL TR-11-35. Vicksburg, MS: U.S. Army Engineer Research and Development Center.

Gingold, R. A., and J. J. Monaghan. 1977. Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 181:375-389.Abaqus 6.5 user's manual (2005).

Guan, P., J. S. Chen, Y. Wu, H. Teng, J. Gaidos, K. Hofstetter, and M. Alsaleh. 2009. A semi-Lagrangian reproducing kernel formulation for modeling earth moving operations. *Mechanics of Materials* 41:670-683.

Guan, P. C., S. W. Chi, J. S. Chen, T. R. Slawson, and M. J. Roth. 2011. Semi-Lagrangian reproducing kernel particle method for fragment-impact problems. *International Journal of Impact Engineering* 38: 1033-1047.

Hughes, T.J.R., and J. Winget. 1980. Finite rotation effects in numerical integration of rate constitutive equations arising in large deformation analysis. *International Journal of Numerical Methods in Engineering* 15(12): 1862-1867.

Lancaster, P., and K. Salkauskas. 1981. Surfaces generated by moving least squares methods. *Mathematics of Computation* 37: 141-158.

Liu, W. K., S. Jun, and Y. F. Zhang. 1995. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids,* 20:1081-1106.

Patran 2010. 2010. MSC Software, Santa Ana, CA.

Puso, M., E. Zywicz, and J. S. Chen. 2006. A new stabilized nodal integration approach. In *Lecture Notes in Computational Science and Engineering* 57:207-218.

Ren, X., J. S. Chen, J. Li, T. R. Slawson, and M. J. Roth. 2011. Micro-cracks informed damage model for brittle solids. *International Journal of Solids and Structures* 48: 1560-1571.

Squillacote, A.H. 2008. *The ParaView Gude, Version 3*. Kitware, Inc. Clifton Park, NY.

Taylor, G. I. 1948. The use of flat-ended projectiles for determining dynamic yield stress, Part I. *Proceedings of the Royal Society of London, Series A* 194:289–299.

Wilkins, M. L., and M. W. Guinan. 1973. Impact of cylinders on rigid boundary. *Journal of Applied Physics* 44:1200–1206.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
December 2012	Final report	

4. TITLE AND SUBTITLE

User's Manual for Nonlinear Meshfree Analysis Program (NMAP) Version 1.0

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

S.W. Chi, C.H. Lee, J.S. Chen, M.J. Roth, and T.R. Slawson

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Engineer Research and Development Center
Geotechnical and Structures Laboratory
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

8. PERFORMING ORGANIZATION REPORT NUMBER

ERDC/GSL TR-12-36

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Engineer Research and Development Center
Geotechnical and Structures Laboratory
3909 Halls Ferry Road
Vicksburg, MS 39180

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

As a part of the meshfree method development for fragment-impact modeling, an enhanced semi-Lagrangian reproducing kernel particle method formulation for modeling large material deformation and damage mechanisms was developed. The formulation includes a microcrack-informed damage model (MIDM) for improved material failure modeling. The MIDM is based on multiscale homogenization using the energy bridging theory pertinent to fragment penetration modeling of concrete materials. Enhanced kernel contact algorithms to model multi-body contact applicable to penetration problems were also developed. The computational formulations and numerical algorithms for these new meshfree method developments were implemented into the parallel Nonlinear Meshfree Analysis Program (NMAP) code. This report provides user instructions for building and running a model in NMAP. The report, together with previously published technical reports, also provides a general overview of the theoretical foundation of the newly developed meshfree formulation for fragment-impact modeling.

15. SUBJECT TERMS

NMAP code
RKPM
meshfree
penetration
user's manual
multiscale

16. SECURITY CLASSIFICATION OF:			**17.LIMITATION OF ABSTRACT**	**18.NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED		86	